



PRIME HERO ACADEMY

Versão 2.1 - Março/2022



Empresa participante do
SCALE UP
ENDEAVOR



TERMO DE CONFIDENCIALIDADE



Este documento contém informações confidenciais e destinam-se unicamente ao uso pelo (s) indivíduo (s) a quem são endereçados ou direcionados através do Programa Prime Experts da Prime Control.

É expressamente proibido qualquer reprodução total ou parcial, compartilhamento ou uso impróprio deste conteúdo sem autorização prévia por escrito da Prime Control.



PrimeControl
Quality Drives Results

AGENDA

AGENDA



- **Nossa Equipe**
- **Fundamentos de Teste**
 - Um pouco de história, conhecendo a área de qualidade de software e sua evolução
 - Fundamentos do Teste
 - Teste de Software
 - O Processo de Testes
 - Técnicas de Modelagem de Teste
 - Ferramentas de Teste
- **Automação de testes com Robot Framework**
 - Configurando ambiente
 - Introdução do Robot Framework
 - Robot Web na prática
 - Um pouco de automação Mobile:



NOSSA EQUIPE



ALDEMIR SALES

Experiência profissional superior há 13 anos em empresas nacionais e multinacionais, atuando nas áreas de tecnologia da informação, serviços e operações, em posições de liderança voltadas para estratégia, organização, gestão e controle para a área de testes. Responsável por projetos de alto impacto para grandes corporações como OI, Grupo Claro, Natura, PagSeguro, Cielo, Santander, BV Financeira, Banco Original, C&A, Burger King, Itaipu, Rede Globo, entre outros. Instrutor há mais de 10 anos para cursos de certificação do ISTQB/BSTQB, já tendo treinado mais de 120 turmas.

Certificações:

- CTFL – Certified Tester Foundation Level, 2011.
- CTAL-TA – Certified Tester Advanced Level – Test Analyst, 2011.
- CTAL-TM – Certified Tester Advanced Level – Test Manager, 2012.



NOSSA EQUIPE



CAIO SKROCH

Carreira desenvolvida na área da Tecnologia da Informação, mais de 14 anos de experiência no desenvolvimento e qualidade de software. Visando garantir a integridade do software em todo o processo produtivo, se utilizando de metodologias ágeis, melhorias contínuas e testes automatizados. Atuando em diversos segmentos, como: bancário, seguros, logística, varejo, alimentício, aviação, food, fiscalização de trânsito e game.



NOSSA EQUIPE

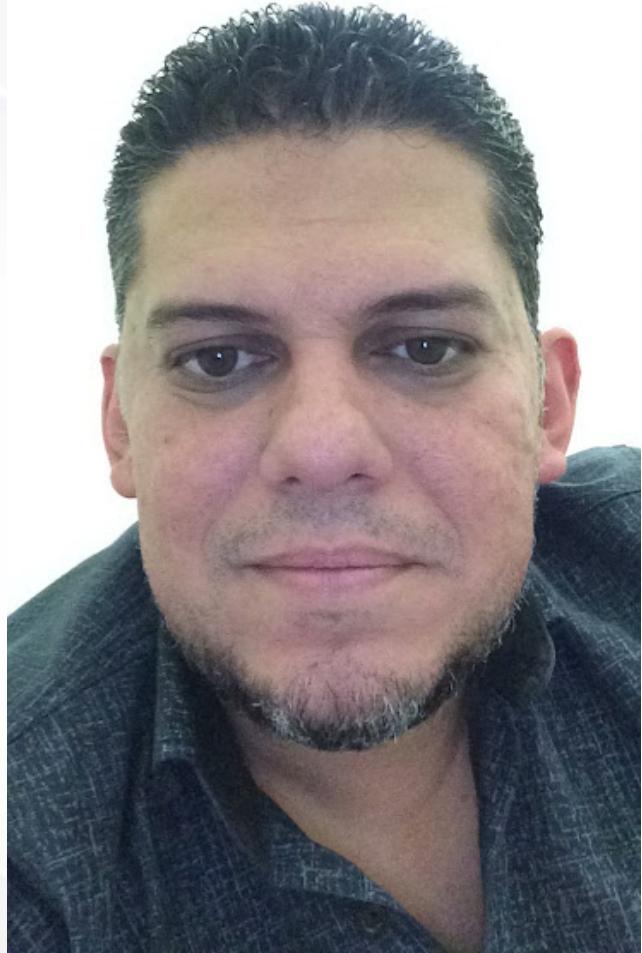


HELDER FERNANDES

Atuante na área de tecnologia a mais de 8 anos em empresas nacionais e multinacionais. Boa parte desta atuação voltada para a área de Quality Assurance com diversas especializações em Automação de testes. Experiências com clientes de diversos nichos de mercado e atuação em diversas pontas do processo de QA como testes de aceitação, testes de performance, API, contrato, mainframe, automação móible e Web.



NOSSA EQUIPE



Ezequiel Lima (Zeq)

- ❑ Aqui na Prime Control sou Tech Lead.
- ❑ Tenho aproximadamente 20 anos de experiência com Testes de Software.
- ❑ Possuo as certificações:
 - ✓ **CTFL** – Certified Tester Foundation Level
 - ✓ **CTAL-TA** - Certified Tester Advanced Level – Test Analyst
 - ✓ **CPRE-FL** – Certified Professional for Requirement Engineering - Foundation Level
- ❑ Ministro treinamentos de:
 - ✓ Formação em Teste de Software a pelo menos 10 anos;
 - ✓ Preparatório para as certificações do ISTQB e IREB;
 - ✓ Gestão (Planejamento, Estimativa, Estratégias, Desenvolvimento de Equipes)
 - ✓ Ágil
 - ✓ Comportamental/Motivacional.



NOSSA EQUIPE



Rodrigo Costa

Experiência profissional de 16 anos em TI, atuando como desenvolvedor, analista de sistemas/testes, coordenador e especialista. Conhecimentos em diversas tecnologias, linguagens de programação e frameworks de testes. Nos últimos 6 anos atuando com foco em disciplinas de testes, testes de performance, automações de páginas web, aplicativos desktop/mobile e webservices. Principais atuações: Itaú Unibanco, Via Varejo, Santander, Easynvest(Nu Investimentos), Guide Investimentos, Unimed Do Brasil, Unimed Seguros, Vivo, Nextel, Tenda Construtora, MC1, Sascar, Americanas e Prime Control.



NOSSA EQUIPE



William Filho

Há pouco mais de 04 anos atuando na área de QA, no momento como Test Lead, com grande expertise gerenciamento de times de alta performance. Atuante em projetos nacionais e internacionais das áreas de varejo, gestão pública, health techs, fintechs e gestão de ativos. Sempre buscando as melhores práticas e as inovações da área de QA para colaborar com sucesso do produto. Conhecimento vasto em metodologias e frameworks de desenvolvimento como também frameworks de automação de testes, seja mobile, desktop, web ou api's e seus respectivos pipelines.



NOSSA EQUIPE



José Cristiano

Experiência profissional superior há 8 anos em empresas nacionais e multinacionais, atuando nas áreas de tecnologia da informação, serviços e operações, em posições de Analista de Teste / Automação de teste, voltado para planejamento de testes, análise, modelagem, análise de risco, implementação, execução dos testes, verificação da qualidade do software e automação de testes. Desempenhando atividade de Analista de testes em projetos de alto impacto para grandes corporações como OI, Natura, Porto Seguro, Lopes, Riachuelo, Super Digital entre outros.

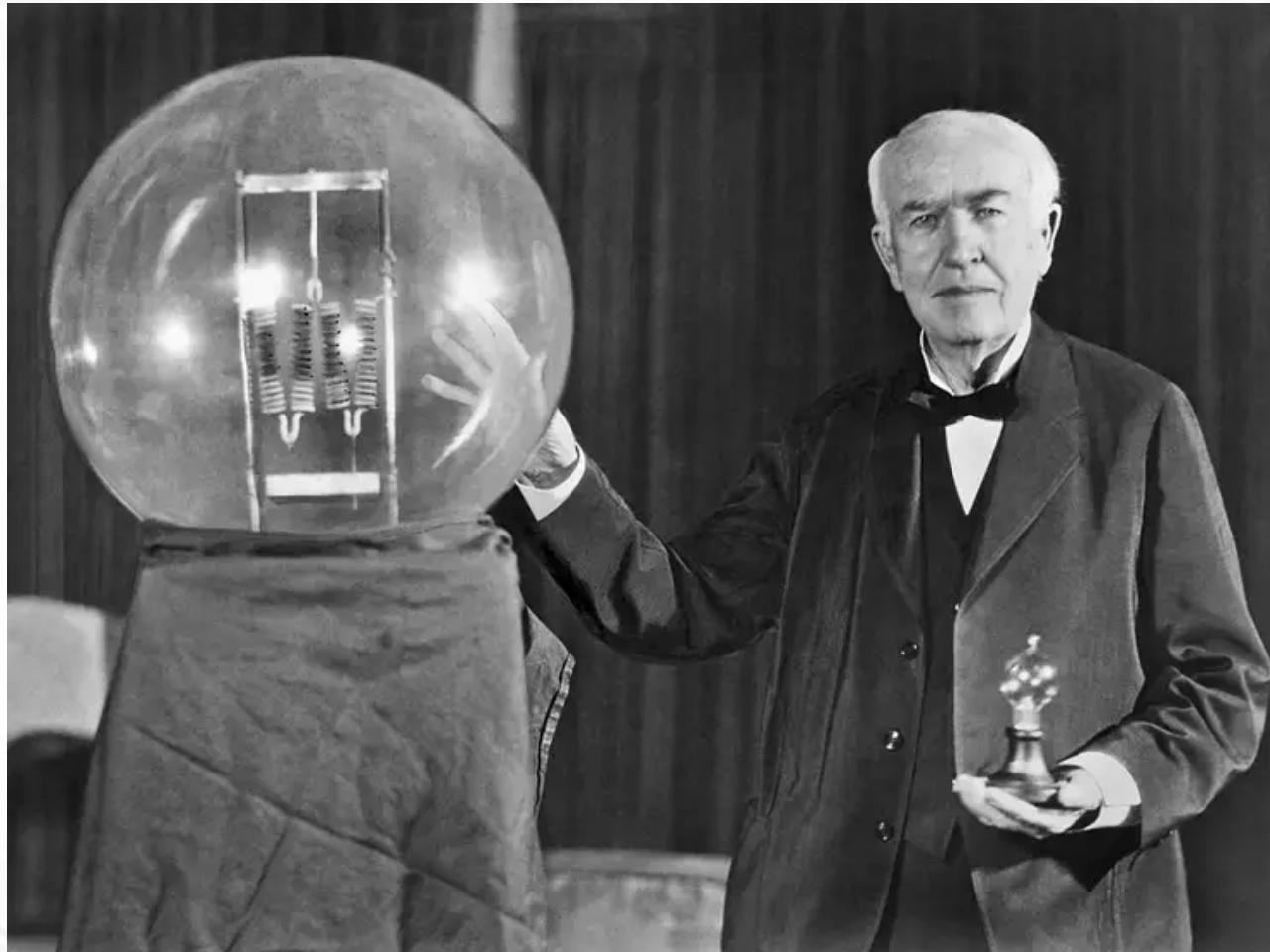


PrimeControl
Quality Drives Results

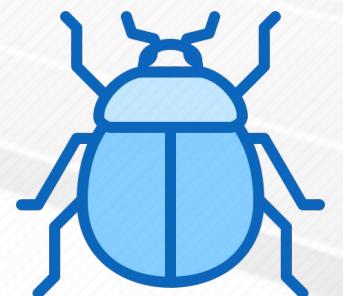
UM POUCO DE HISTÓRIA



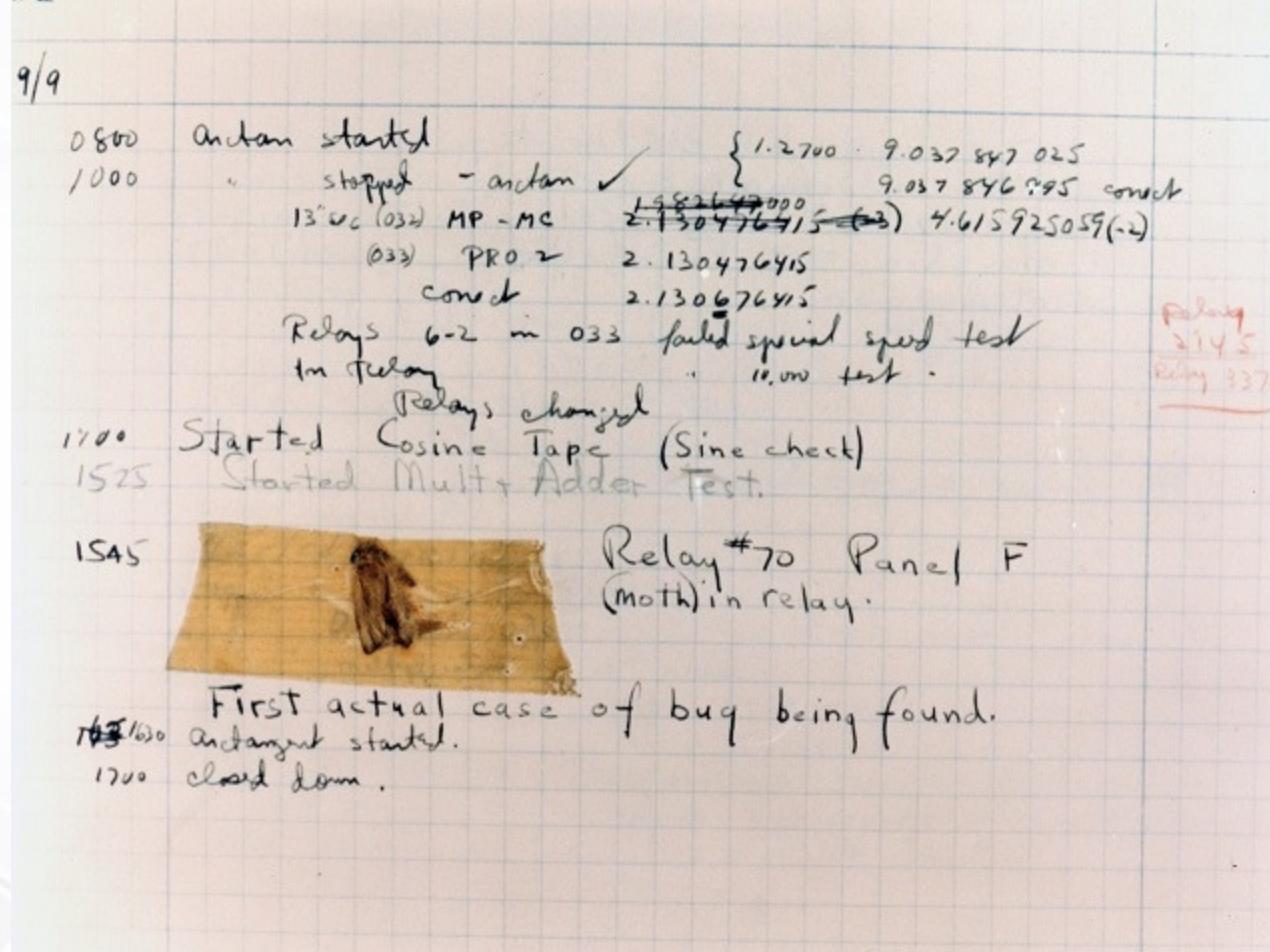
1878



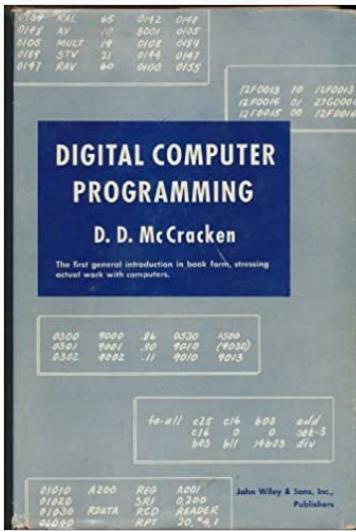
Thomas Edson



O Primeiro “BUG” de Computadores

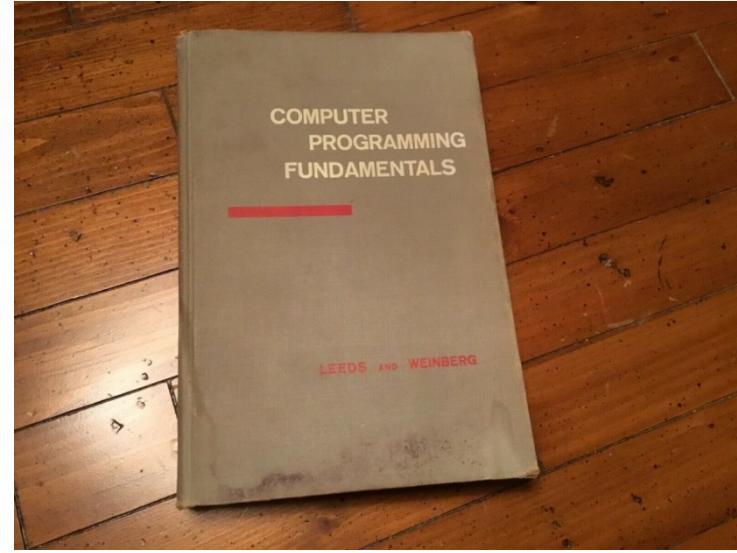


1947



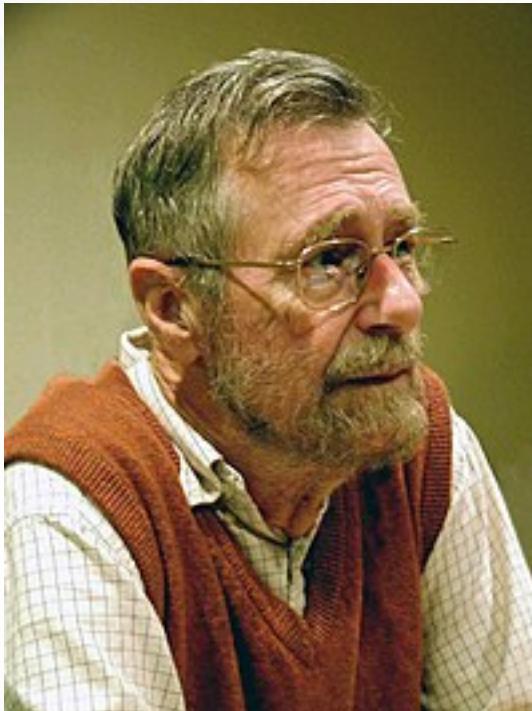
Revisando esse livro, o Engenheiro de Software Charles L. Baker estabelece a diferença entre tirar defeitos e testar software

1957



Primeiro livro a conter um capítulo apresenta capítulo sobre teste de software.

1961

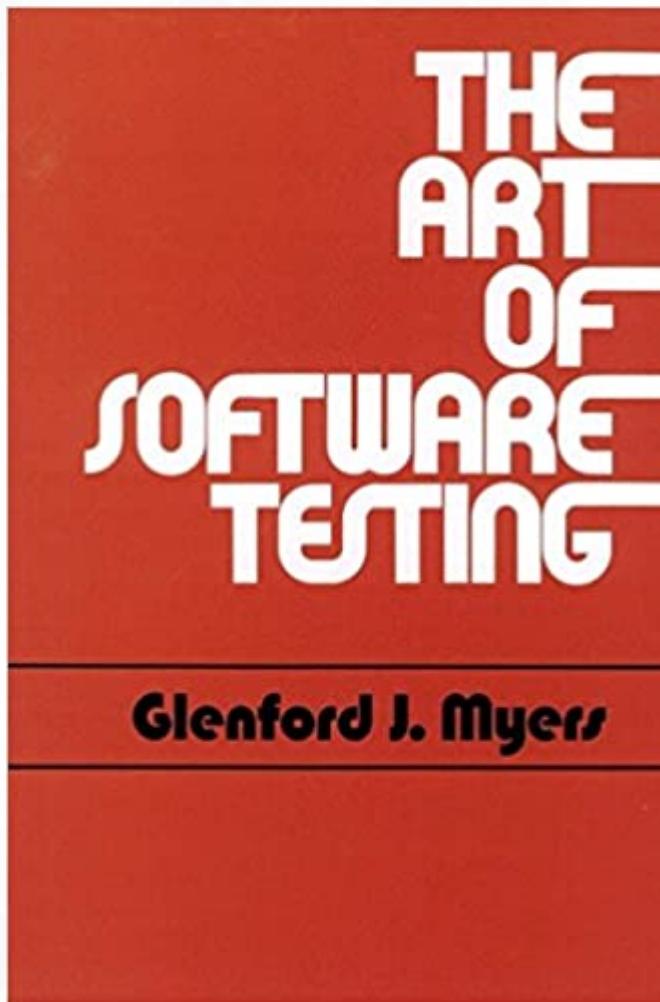
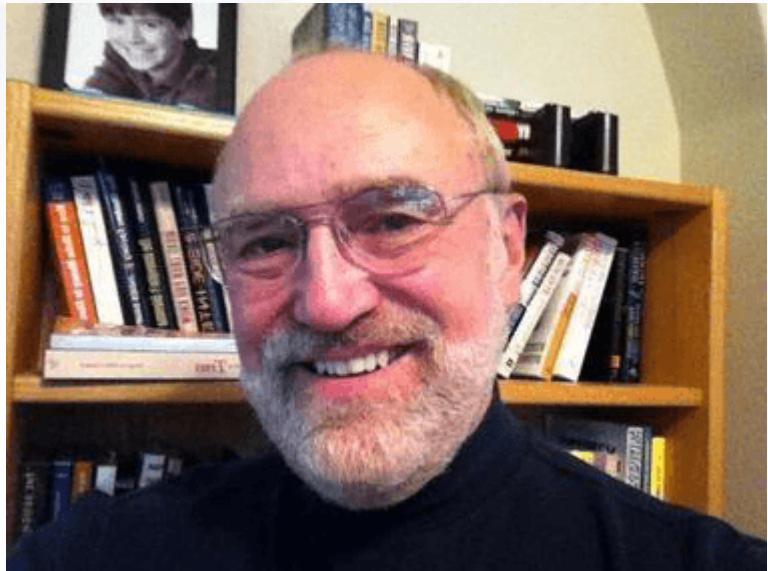


“ O teste mostra a
presença, não a ausência
de erros”

Dijkstra



A citação famosa de Dijkstra teria sido falada pela primeira vez uma conferência do Comitê Científico da OTAN, na Itália, outubro de 1969.



Glenford Myers publica o livro *The Art of Software Testing*, que apresentou diversas ideias e conceitos inovadores sobre a atividade de teste de software para a época. A data de lançamento da obra também deu origem ao dia internacional do teste de software, 20 de fevereiro de 1979.

1979



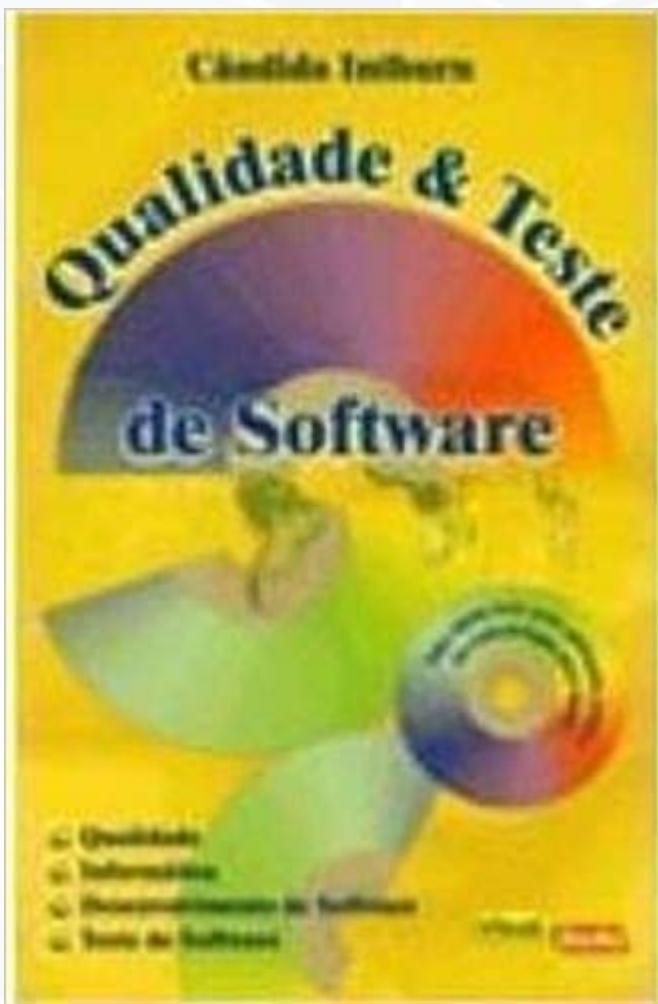
*Advancing Technology
for Humanity*

The world's largest professional association
for the advancement of technology

<https://www.ieee.org>

O Instituto de Engenheiros Eletricistas e Eletrônicos, chamado IEEE, organização americana, mas que influencia o desenvolvimento tecnológico todo o mundo, publica a primeira versão do padrão de documentação de teste de software.

1983



<https://www.istqb.org>

➤ 2001



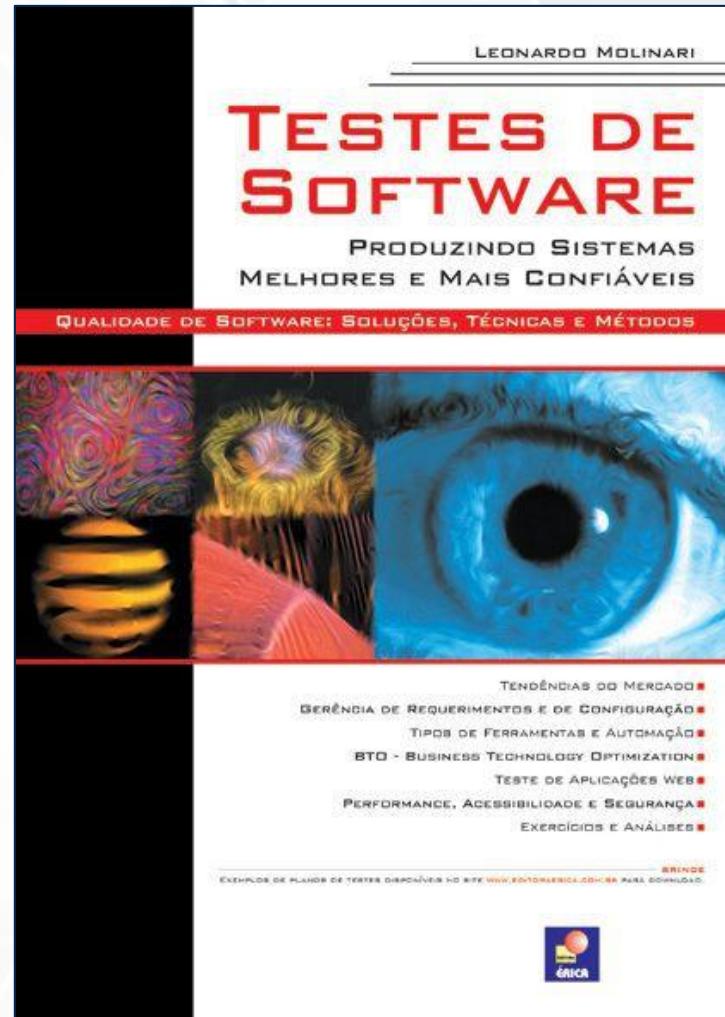
Teste de Software

3^a Edição revisada e ampliada



Emerson Rios
Trayahú Moreira

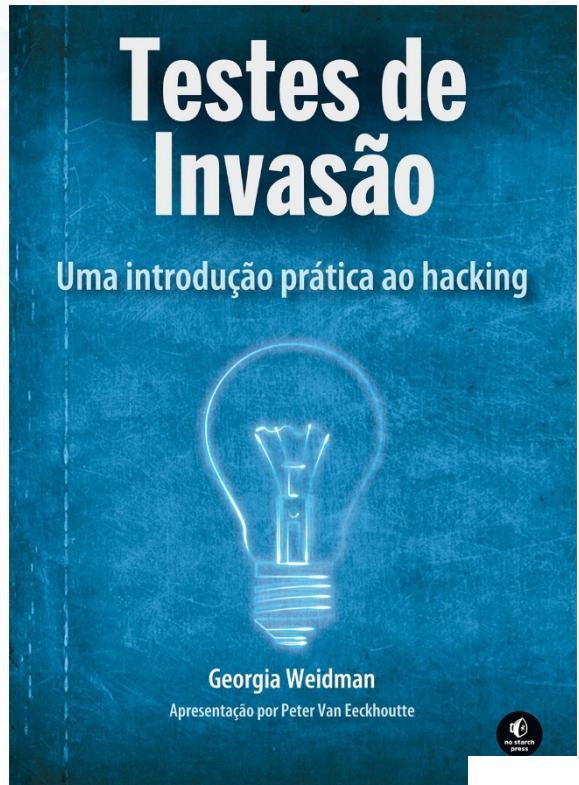
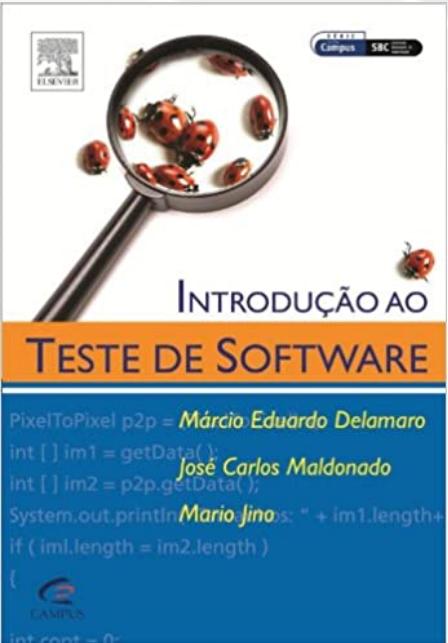
2003





O núcleo da ferramenta Selenium, muito popular para automatizar aplicações web para fins de teste, é desenvolvido pela empresa ThoughtWorks Chicago.

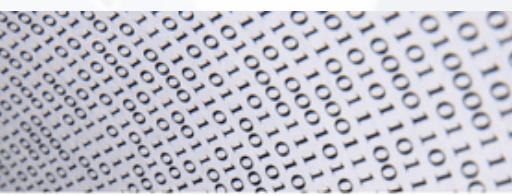
2004



CINTEQ

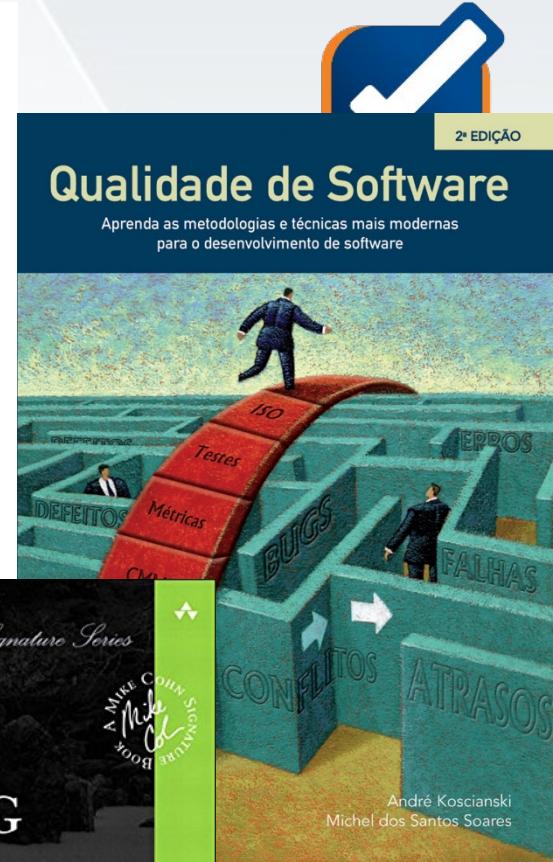
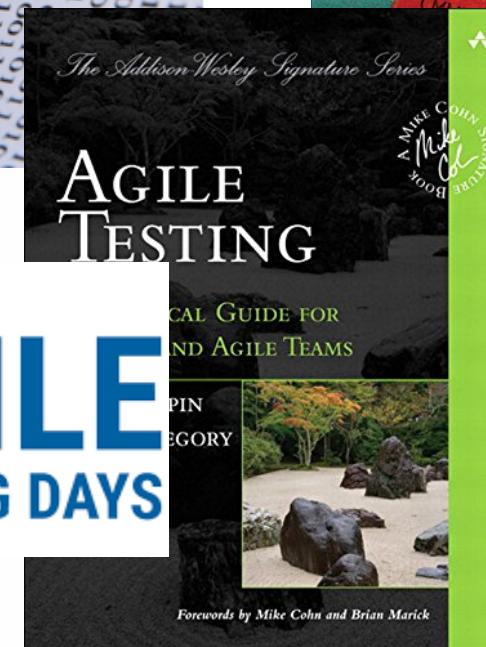


AGILE
TESTING DAYS



Base de conhecimento em teste de software

Aderson Bastos Emerson Rios Ricardo Cristalli Trayahú Moreira





E HOJE EM DIA?



QUAL O PAPEL DO QA NO TIME?

Na minha máquina funciona!



QUAL O PAPEL DO QA NO TIME?



O papel do QA 3.0 se resume a 3 skills que ele deve possuir:

- Desenvolvimento
- Negócios
- DevOps

No nosso mundo o QA não é um papel apartado no time, ele deve ser integrante do time de desenvolvimento!



DESENVOLVIMENTO

Tá bom, mas porque eu preciso saber desenvolvimento pra ser QA? O que isso influênciaria no meu dia a dia?



NEGÓCIOS

BDD, critérios de aceite, regras de negócio, planejamento, entregas, prazos e amor pelo produto.



DEVOPS

Continuous Integration, Jenkins, GitLab Ci, Circle Ci, Farms, AWS.



**SEJA UM
ESPECIALISTA!**

POR QUE ESPECIALISTA?



O QA deve ser o especialista quando o assunto é Qualidade, ou seja, ele deve saber indicar qual stack de ferramentas o cliente deve usar de acordo com seu desenvolvimento. O QA deve representar a área de qualidade no projeto e ser referência em todos os aspectos relacionados.

1. Indicar ferramentas
2. Sugerir melhorias
3. Implantar melhorias
4. Disseminar a cultura ágil
5. Disseminar a cultura de QA

O QUE ESTUDAR?



Desenvolvimento

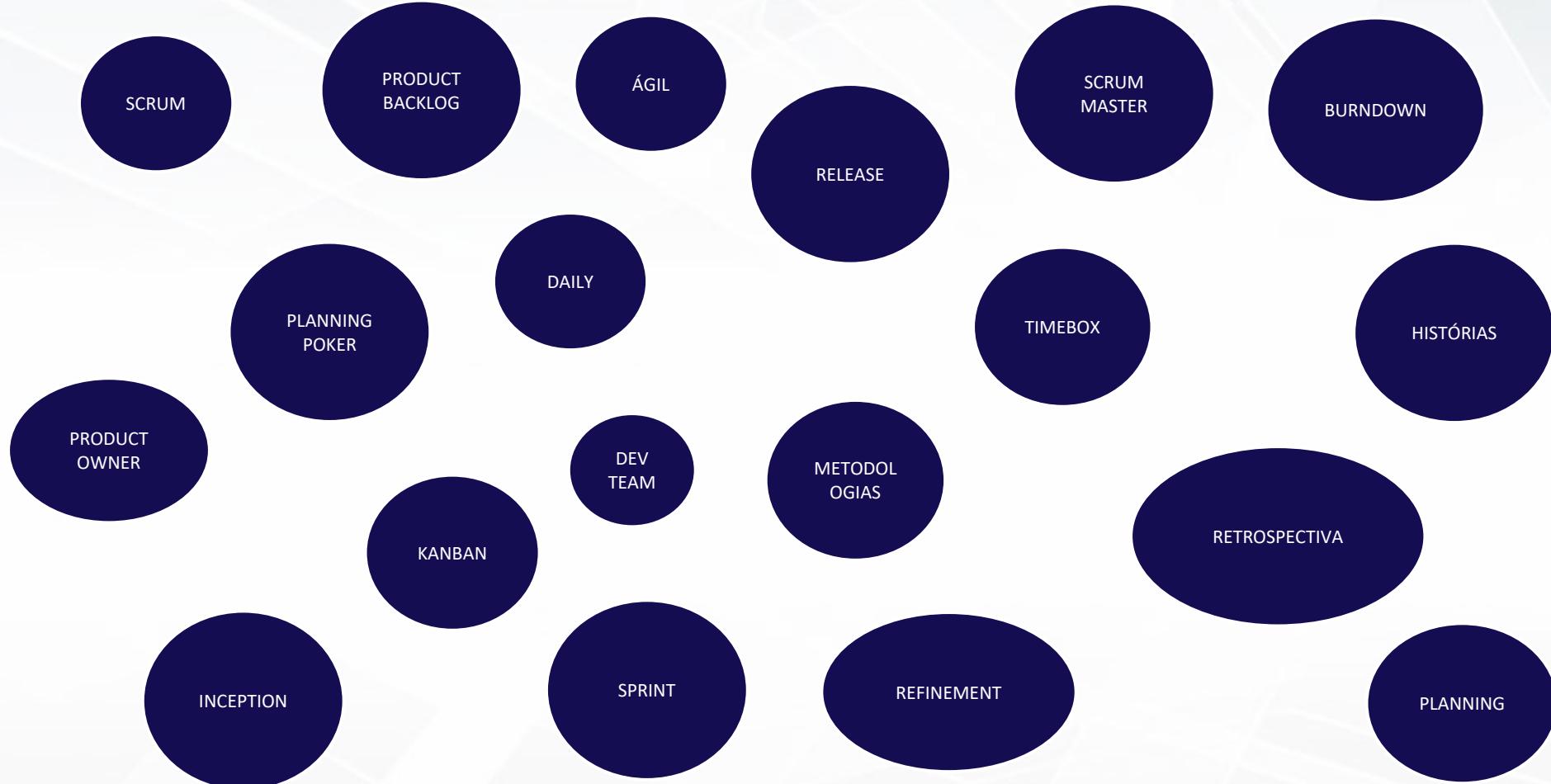
- Lógica de Programação
- Orientação a Objetos
- Dominar uma Linguagem de Programação
- Conhecimento em API's
- Ciclo de desenvolvimento de um projeto
- Banco de dados

Negócios

- Regras de negócio
- Critérios de aceite
- Escrita de cenários
- Sentimento de dono

DevOps

- Integração contínua
- Automação de tarefas
- Versionamento
- Implantação
- Infra
- Aws
- Farms





E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





PrimeControl
Quality Drives Results

Requisitos....
... A origem de tudo

Engenharia de Requisitos



O que é?

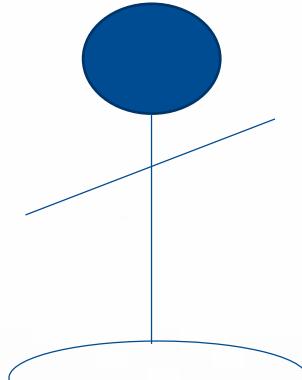
A **engenharia de requisitos** é um processo que engloba todas as atividades que contribuem para a produção de um documento de **requisitos** e sua manutenção ao longo do tempo.

Engenharia de Requisitos



1. Tudo começa quando há uma necessidade que precisa ser atendida...

Analista de Requisitos



Alguém precisa de algo novo...

Sistemas precisam de modificações...

As tecnologias evoluem, ficam obsoletas...

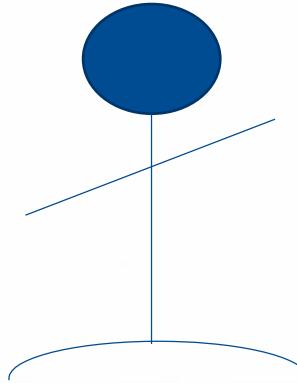
Engenharia de Requisitos



2. O Analista de Requisitos ouve e anota as necessidades....

Esta atividade é chamada de Levantamento de Requisitos.

Analista de Requisitos



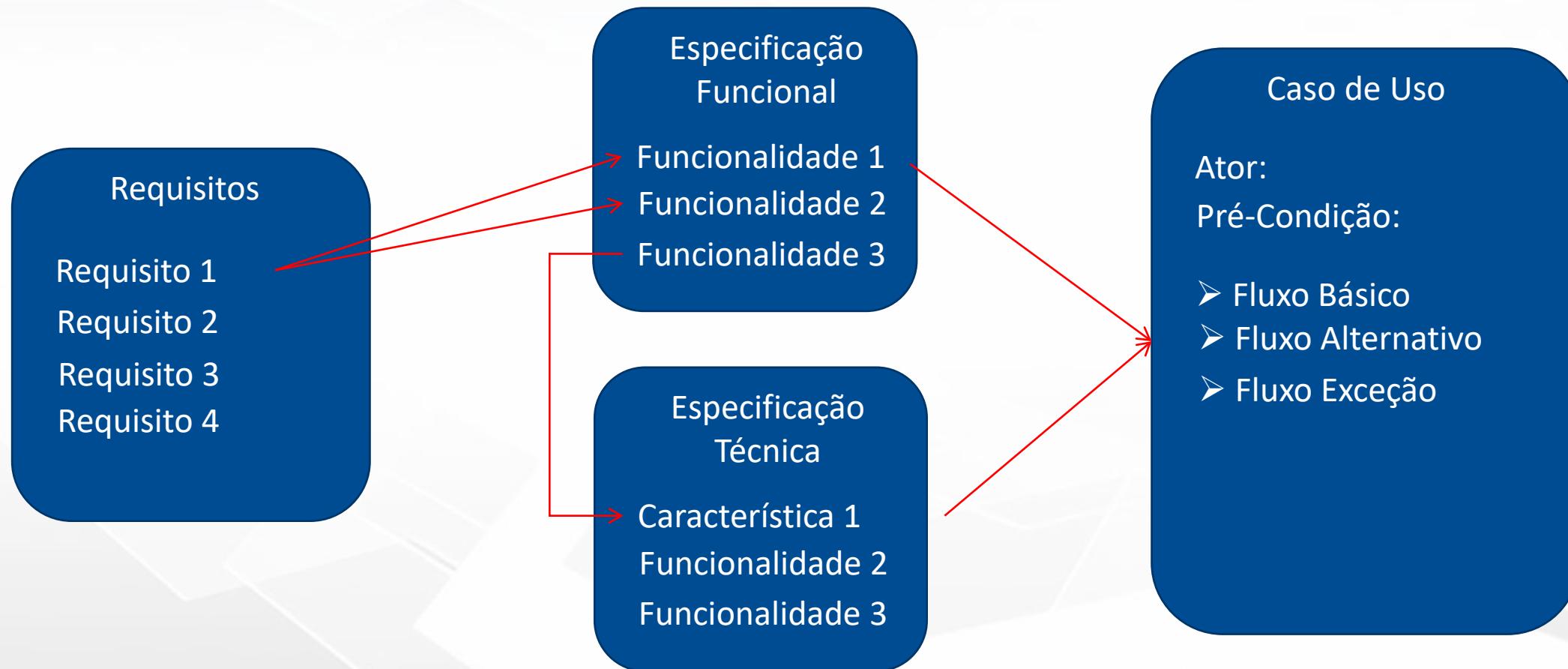
Normalmente esse levantamento acontece por meio de entrevistas

Esta atividade resulta em um documento com as necessidades do usuário estruturada.

Engenharia de Requisitos



3. Depois de levantado, agora é preciso analisar, estruturar, dar respostas e soluções sistêmicas para esses requisitos....





PrimeControl
Quality Drives Results

FUNDAMENTOS DO TESTE

O QUE VAMOS VER?



- Objetivos do teste
- Características da qualidade
- Tipos de teste
- Verificação e validação de teste
- Pirâmide de teste

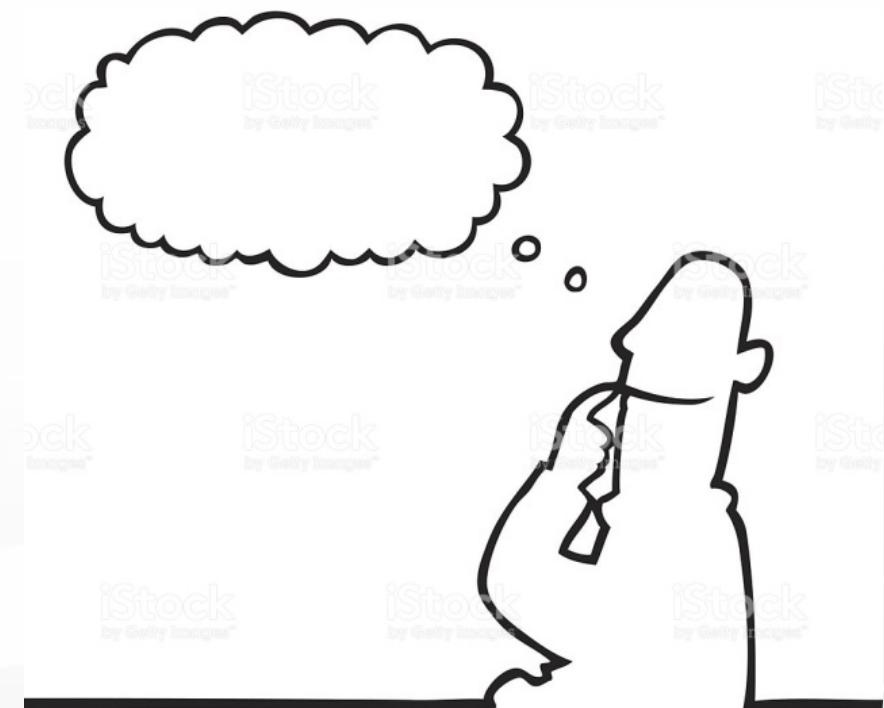


OBJETIVOS DOS TESTES

O QUE É TESTE DE SOFTWARE?



- Teste **não** significa documentação
- Teste é o que você **pensa e faz**
 - Foco em processo e produto



OBJETIVOS DO TESTE DE SOFTWARE



Encontrar **defeito**, o mais **cedo** possível, e **garantir** que sejam corrigidos
[Patton, 2005]

- O que é um bug no software?
- Como um bug pode ser encontrado?
- **Contribuir** com a qualidade do produto
 - O que é **qualidade**?
- Teste **NÃO** mostram ausência de bugs
- Qual a relação de teste com qualidade?

BUG NO SOFTWARE



O software tem um bug quando:

- **Não** esta conforme com a especificação
- Faz algo **não** especificado
- Não faz algo que deveria fazer, mas **não** está especificado
- É difícil de usar, entender
- É lento, não responsivo
- Apresenta riscos ao usuário
- **Não** representa o que o cliente quer

TESTE DE CONCENTRAÇÃO



Quantas vezes a bola bate no chão?

<https://www.youtube.com/watch?v=4xC9uoAd4Pk>

A IMPORTÂNCIA DO TESTE DE SOFTWARE



- Por que testes são importantes?
- Dá para passar sem eles?
- Testar é caro?
- Bug é caro?

A IMPORTÂNCIA DO TESTE DE SOFTWARE



Tragédia do voo AF 447

Conclusões do 3º relatório do Escritório de Investigações e Análises da França (BEA)



*Estol – situação de perda de sustentação

Situação dos corpos resgatados

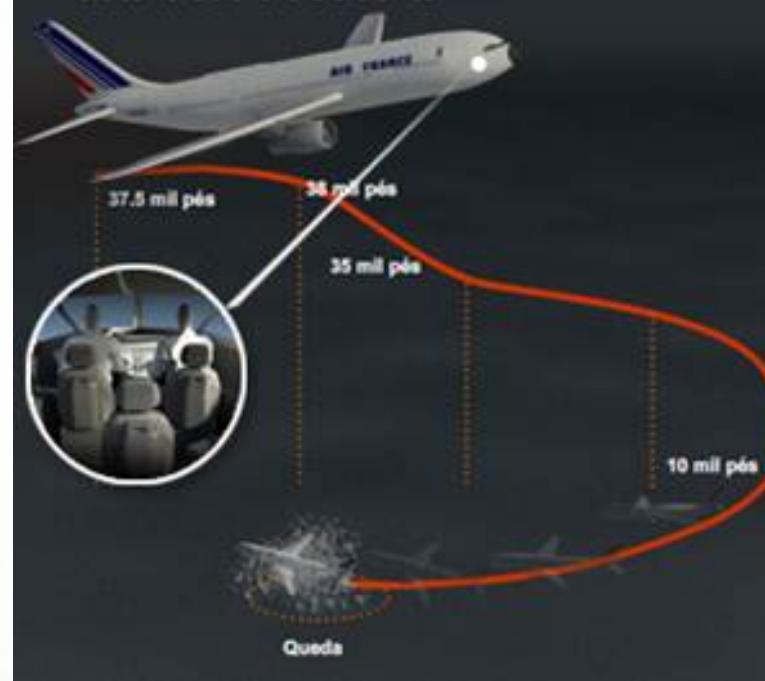
154 corpos foram retirados do mar

50 estavam flutuando no oceano em 31 de maio de 2009



104 restos mortais foram resgatados e ainda precisam ser identificados

Últimos minutos do voo AF 447 3º relatório sobre o acidente



<https://www.youtube.com/watch?v=GJE0E72jC1o>

A IMPORTÂNCIA DO TESTE DE SOFTWARE



Show de Madonna: fãs que tiveram cartão debitado terão ingresso

O anúncio afirma que o motivo da **pane no sistema** no site ticketsforfun.com.br no início das vendas dos ingressos (dia 1º/9 no Rio e 3/9 em São Paulo) foi devido ao "número de hits simultâneos no site de vendas [que] provocou morosidade no sistema. Como efeito colateral, houve **aumento do tempo de transação** entre os diversos módulos do software de vendas causando **atraso nas respostas** e gerando **informações múltiplas** que o sistema não conseguiu interpretar".



A IMPORTÂNCIA DO TESTE DE SOFTWARE

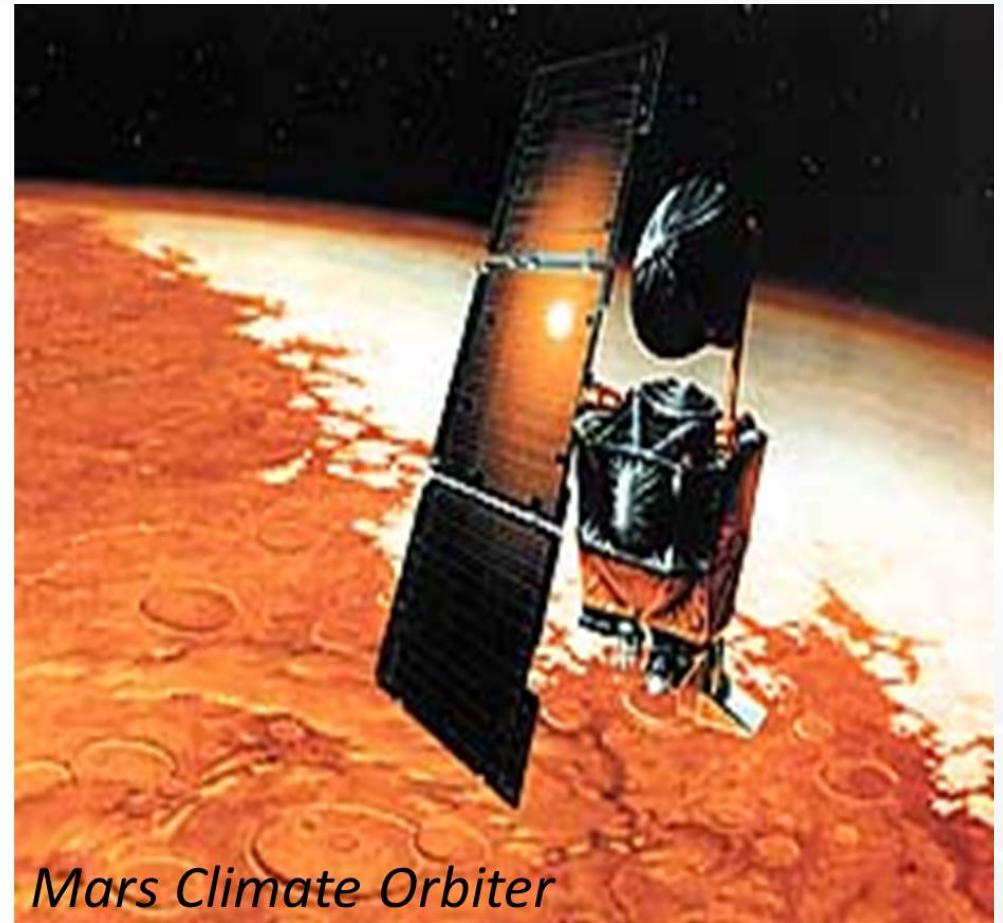


- Blackout atinge norte dos EUA em 2006
- Possível **sobrecarga** em uma das estações de energia
- Prejuízo estimado US\$ 6 bilhões

A IMPORTÂNCIA DO TESTE DE SOFTWARE



- NASA perde sonda que ia para Marte
- Erro na conversão entre unidades de medida dos padrões métricos e imperial
- Prejuízo estimado US\$ 125 milhões

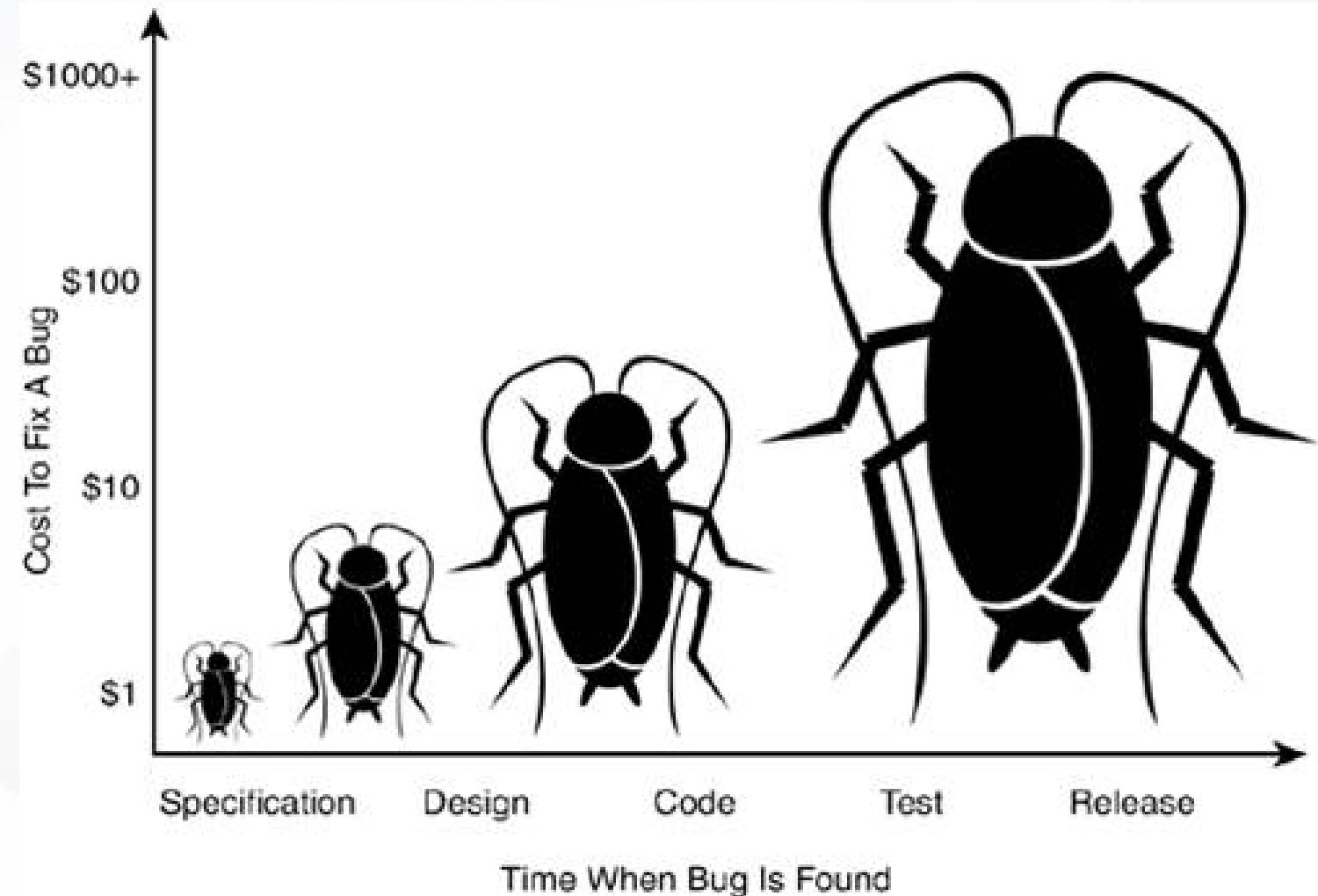


Mars Climate Orbiter

A IMPORTÂNCIA DO TESTE DE SOFTWARE



Regra 10 de Myers





CARACTERÍSTICAS DA QUALIDADE

CARACTERÍSTICAS DA QUALIDADE



A norma internacional ISO/IEC, define qualidade de software como:

“A totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.

CARACTERÍSTICAS DA QUALIDADE



- Necessidades explícitas:
 - São aquelas definidas no requisito proposto. Esses requisitos devem definir as condições em que o produto deve ser utilizado e dizer seus objetivos, funções e desempenho esperado. São, portanto, fatores relativos à qualidade do processo do desenvolvimento do produto que são percebidos somente pelas pessoas que trabalharam no seu desenvolvimento.

CARACTERÍSTICAS DA QUALIDADE



- Necessidades implícitas:
 - São aquelas que, embora não expressas no documento do produtor, são necessárias para o usuário. Estão englobados em esta classe os requisitos que não precisam ser declarados por serem óbvios, mas que pela gravidade de suas consequências devem ser levados em consideração.

CARACTERÍSTICAS DA QUALIDADE



- Freeman (1987) apresenta uma distinção entre qualidade básica e qualidade extra.
 - Em qualidade básica ele lista: funcionalidade, confiabilidade, facilidade de uso, economia e segurança de uso.
 - Em qualidade extra ele lista: flexibilidade, facilidade de reparo, adaptabilidade, facilidade de entendimento, boa documentação e facilidade de adicionar melhorias.



TIPOS DE TESTE

TIPOS DE TESTES DE SOFTWARE



Funcionais

- Teste de unidade
- Teste de integração
- Teste de operação
- Teste Positivo-negativo
- Teste de regressão
- Teste de Usabilidade
- Teste de Aceitação

Não-funcionais

- Teste de performance
- Teste de carga
- Teste volume
- Teste de stress
- Teste de segurança



VERIFICAÇÃO E VALIDAÇÃO

VERIFICAÇÃO E VALIDAÇÃO DE TESTE



As técnicas de verificação e validação são fundamentais para identificar se um software possui defeitos e está de acordo com o especificado.
(BASTOS, Aderson et al., 2007)

VERIFICAÇÃO E VALIDAÇÃO DE TESTE



- A diferença entre verificação e validação é explicada de forma sucinta com as seguintes perguntas (Boehm, 1979):
 - Verificação: estamos construindo certo o produto?
 - Validação: estamos construindo o produto certo?

VERIFICAÇÃO E VALIDAÇÃO DE TESTE



Verificação

- Revisão de requisitos
- Revisão de modelos
- Inspeções de código
- Revisões e inspeções técnicas em geral

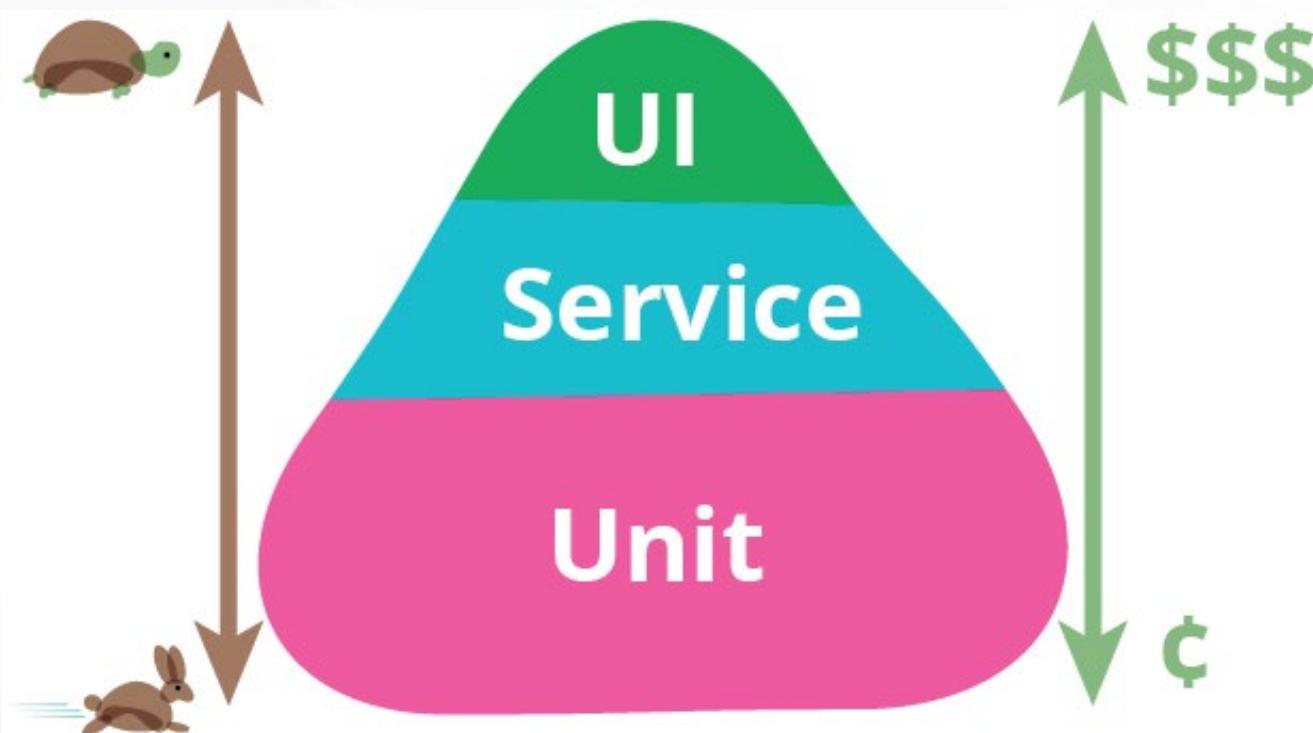
Validação

- Teste unitário
- Teste de integração
- Teste de sistemas
- Teste de aceitação
- Teste de regressão



PIRÂMIDE DE TESTE

PIRÂMIDE DE TESTE



A pirâmide de testes é uma maneira de pensar sobre diferentes tipos de testes automatizados que devem ser usados para criar um portfólio equilibrado. Seu ponto essencial é que você deve ter muito mais testes unitários e pouco teste de tela. (Fowler, 2012)

PIRÂMIDE DE TESTE



Vantagens

- Agilidade na execução dos testes
- Menor tempo de construção do teste
- Não há necessidade de criação de ambiente
- Baixa manutenção de scripts
- Sem necessidade de licença da ferramenta



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





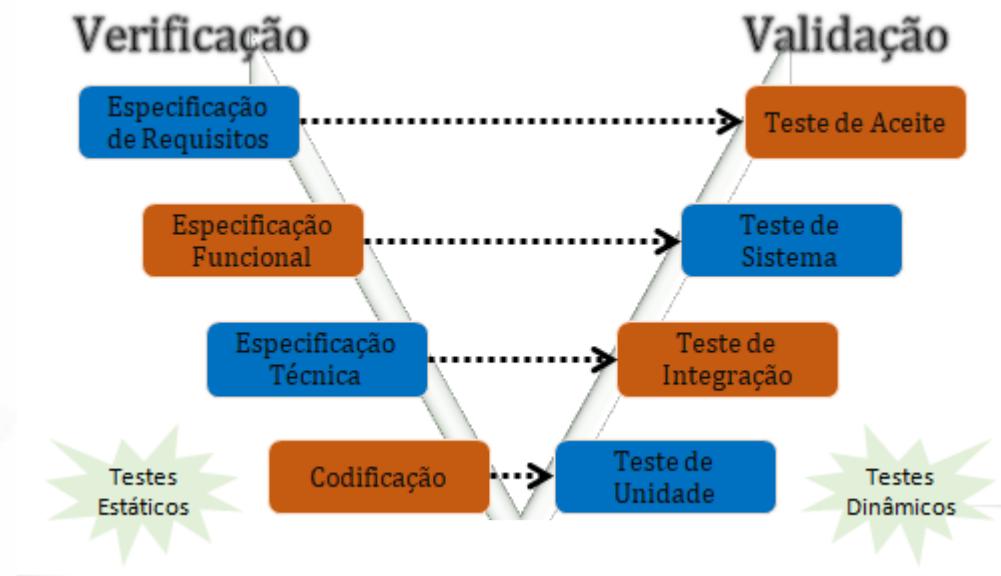
PrimeControl
Quality Drives Results

TESTE DE SOFTWARE

O QUE É TESTE DE SOFTWARE?



Consiste de uma verificação (análise estática) e validação (análise dinâmica) do produto, e é um processo relevante para **Prevenir** que **Erros** se transformem em **Defeitos (Bugs)** e **Defeitos gerem Falhas Sistêmicas**.



*Resumidamente, teste é **verificação** e **validação**, estruturado de maneira metódica por meio de um processo claro e bem definido.*

A IMPORTÂNCIA DO TESTE



Tudo começa com pessoas requisitando e estruturando o requisitado e tudo isso por meio do processo da comunicação e esse processo é cheio de Gap's (telefone sem fio, lembra?) como vimos no capítulo 1.

- Qual é a probabilidade de quem pediu não ter pedido direito?
- Qual é a probabilidade de quem ouviu não ter ouvido direito?
- Qual é a probabilidade de quem escreveu a documentação de requisitos não ter escrito direito?
- Qual é a probabilidade de quem vai programar com base nessas documentações programar com bug?
- Qual é a probabilidade desse sistema apresentar falha depois de pronto?

Certamente as respostas te levarão à concluir que todo desenvolvimento necessita de teste de software pela mesma razão que os lápis necessitam de borrachas....

*Na Psicologia a máxima é: “Penso, logo existo!”
em teste de software é: “Penso, logo erro...”*

Em Deus nós acreditamos, todo resto agente testa...

A IMPORTÂNCIA DO TESTE – CONCEITOS



Erro

Trata-se de uma ação humana.

Ex.: O analista entendeu errado o que o usuário estava falando... Quando ele projetar esse entendimento errado se transforma em bug.

Defeito/Bug

Trata-se de algo evidente, visível, quando o erro é materializado (escrito, programado) passa a ser um defeito.

Ex.: ...deverá imprimir!

Mas aonde? Na tela, PDF, Impressora?

Falha

Resultado de um código defeituoso executado.

Além do erro, o equívoco e o engano também são uma ação humana, no entanto, não geram defeito na documentação, podem gerar defeitos de programação, podem gerar falhas, mas somente o “erro” gera defeito de documentação.

Ex.: “era para apertar a tecla “enter” para prosseguir e o usuário apertou a tecla “backspace” fazendo retroceder o sistema.

QUALIDADE DO SOFTWARE



Qualidade é ...

- estar em conformidade com os requisitos dos clientes
- antecipar e satisfazer os desejos dos clientes
- escrever tudo o que se deve fazer e fazer tudo o que foi escrito

A totalidade das características de uma entidade
que lhe confere a capacidade de satisfazer
às necessidades **explícitas** e **implícitas**

QUALIDADE DO SOFTWARE



Teste e qualidade

- ✓ Teste fornece confiança quando encontra poucos bugs
 - ✓ Passar no teste reduz o nível do risco de qualidade
 - ✓ Encontrar bugs dá a chance de aumentar a qualidade
 - ✓ O conjunto de teste dá uma avaliação da qualidade
-
- O teste é **parte de uma estratégia maior** de garantia da qualidade para um projeto
 - O teste **provê informação** para guiar o projeto, reduzir e gerenciar os riscos, e reparar os problemas importantes
 - O teste também pode corresponder às necessidades de conformidade, contratuais, e regulamentares

QUALIDADE DO SOFTWARE



Garantir a qualidade é ...

- ✓ Estabelecer processos e modelos que tornem o resultado do produto de software previsível
- ✓ Estabelecer o processo de teste
- ✓ Estabelecer, colher e analisar métricas de produtividade, eficiência e defeitos nos processos de desenvolvimento e teste de software
- ✓ Estudar e aplicar modelos e padrões reconhecidos para alcançar níveis nacionais ou internacionais de qualidade

É um conjunto de atividades que assegura que todos os esforços serão feitos para que os produtos de software tenham a qualidade desejada

QUALIDADE DO SOFTWARE



*A atividade de teste de software é **um elemento crítico** da garantia de qualidade de software e representa a última revisão de especificação, projeto e codificação*

Pressman

Roger S. Pressman é um engenheiro de software, escritor e consultor, norte-americano, presidente da R.S. Pressman & Associates.

Roger S. Pressman é uma autoridade reconhecida internacionalmente nas tecnologias em melhoria de processos de software e engenharia de software.

PERFIL DO TESTER



O teste é uma atividade específica que requer conhecimento específico

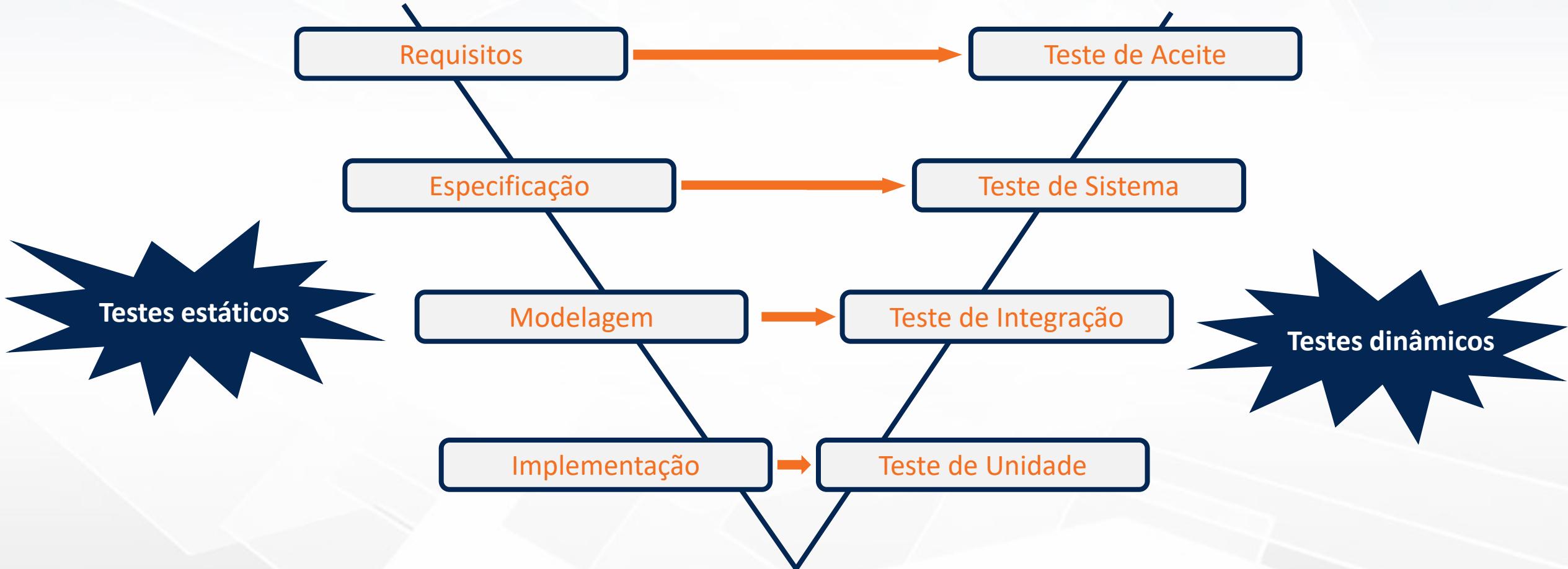
- Perfil do profissional
 - Detalhista
 - Perfeccionista
 - Criativo
 - Organizado
 - Aprendizado Contínuo

EQUIPE DE TRADICIONAL DE TESTES



- ✓ Líder de Teste – Responsável pela liderança de um projeto de teste específico
- ✓ Arquiteto de Teste – Responsável pela montagem do ambiente de teste (infra-estrutura) e escolha de ferramentas
- ✓ Analista de Teste (Engenheiro de Teste) – Responsável pela modelagem e elaboração dos casos de testes e scripts de teste
- ✓ Testador – Responsável pela execução dos casos de testes e scripts de teste

ABORDAGEM DO TESTE



ABORDAGEM DO TESTE – DINÂMICO



Teste de Aceite

Avalia a qualidade e fornece informação sobre o risco da entrega/liberação. Verifica se o sistema atende aos requisitos, como especificados em contrato, do ponto de vista do cliente

Teste de Sistema

Verifica se o sistema como um todo atende aos requisitos especificados. Encontra bugs no comportamento geral através de teste funcional e não-funcional

Teste de Integração

Verifica se grupos de componentes colaboram na forma em que foram especificados pelo projeto técnico do sistema, encontrando bugs nos relacionamentos e interfaces entre pares e grupos de componentes

Teste de Unidade

Encontra bugs nas partes individuais do sistema sob teste. Verifica se cada componente de software executa corretamente de acordo com sua especificação

ABORDAGEM DO TESTE – DINÂMICO



Teste de unidade

Uma unidade é

- Um elemento que possa ser logicamente tratado
- O menor item testável independentemente

O objetivo do **teste de unidade** é encontrar bugs, adquirir confiança, e reduzir os riscos nas partes individuais do sistema sob teste

O responsável por essa fase de teste é geralmente o **desenvolvedor**, visto que:

- ...envolve acesso ao código
- ...é executado no ambiente de desenvolvimento

Tipos de teste

- Funcionais
- Uso de recursos
- Desempenho

ABORDAGEM DO TESTE – DINÂMICO



Teste de integração

O objetivo do **teste de integração** é validar basicamente se as unidades testadas de forma individual executam corretamente suas funções quando colocadas juntas, isto é, quando integradas

Os responsáveis por essa fase de teste são **desenvolvedores e testadores**

- A técnica de integração ideal deve ser analisada

Tipos de teste

- Funcionais
- Uso de recursos
- Desempenho

ABORDAGEM DO TESTE – DINÂMICO



Teste de sistema

O objetivo do teste de sistema é encontrar bugs, adquirir confiança, e reduzir riscos nos comportamentos global e particular, funções, e respostas do sistema sob teste como um todo

Os responsáveis por essa fase de teste são tipicamente **testadores** independentes

O sistema está completo e o ambiente de teste deve ser o mais realístico possível

Tipos de teste

- Funcionais
- Não-funcionais
 - Desempenho, Confiabilidade, Usabilidade, Portabilidade

ABORDAGEM DO TESTE – DINÂMICO



Teste de aceite

O objetivo do teste de aceite é demonstrar que o produto está pronto para entrega/lançamento

Responsáveis por essa fase de teste são geralmente **usuários** ou **clientes**, mas também testadores independentes

- Um teste de aceite pode variar de um test drive informal a uma série de testes planejados e sistematicamente executados

Podemos nos basear em

- Requisitos, contratos, experiência

Tipos de teste

- Funcionais, desempenho, usabilidade

ABORDAGEM DO TESTE – DINÂMICO



Teste de confirmação

Quando um defeito é detectado e resolvido, o software pode ser re-testado para confirmar que o defeito inicial foi removido.

As mudanças devem ser re-testadas para confirmar que a correção foi realizada com sucesso (teste de confirmação). Mas como há riscos de acontecer efeitos colaterais devido a essas mudanças, mesmo em partes não alteradas, os casos de teste são repetidos (testes de regressão).

Re-teste confirma que...

- Mudanças feitas no sistema estão presentes
- Correções de bugs introduzidas no sistema resolveram os sintomas observados

ABORDAGEM DO TESTE – DINÂMICO



Teste de regressão

A cada novo módulo adicionado ou alteração, o software se modifica. Essas modificações podem introduzir novos defeitos, inclusive em funções que previamente funcionavam corretamente

Assim, é necessário verificar se as alterações efetuadas estão corretas e, portanto, deve-se re-executar algum subconjunto de testes que já foi conduzido para garantir que as modificações não estão propagando efeitos colaterais indesejados (Pressman, 2006)

ABORDAGEM DO TESTE – DINÂMICO



Teste de regressão

A cada novo módulo adicionado ou alteração, o software se modifica. Essas modificações podem introduzir novos defeitos, inclusive em funções que previamente funcionavam corretamente

Assim, é necessário verificar se as alterações efetuadas estão corretas e, portanto, deve-se re-executar algum subconjunto de testes que já foi conduzido para garantir que as modificações não estão propagando efeitos colaterais indesejados (Pressman, 2006)

Estratégias:

- Repetição de um conjunto de testes já realizado
 - visa garantir que mudanças no software não introduzam efeitos indesejados ou erros adicionais
 - automatização, auxilia cobertura em profundidade
- 3 categorias:
 - Um conjunto representativo para testar todas as funções do software
 - Testes para as funções mais afetadas pelas alterações
 - Testes para os componentes que sofreram modificações



TIPOS DE TESTE

ISO/IEC
25010 antiga
ISO 9126

Características / Subcaracterísticas:

Funcionalidade: Conveniência, precisão, interoperabilidade, segurança.

Confiabilidade: maturidade (robustez), tolerância a falhas, capacidade de recuperação.

Usabilidade: facilidade de entendimento, capacidade de aprendizagem, operação, atratividade.

Eficiência: comportamento temporal, utilização de recursos.

Manutenibilidade: facilidade de análise, possibilidade de efetuar mudanças, estabilidade, testabilidade.

Portabilidade: adaptabilidade, instalabilidade, coexistência, substituição.

Alvo de teste funcional

Alvo de teste não-funcional



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA

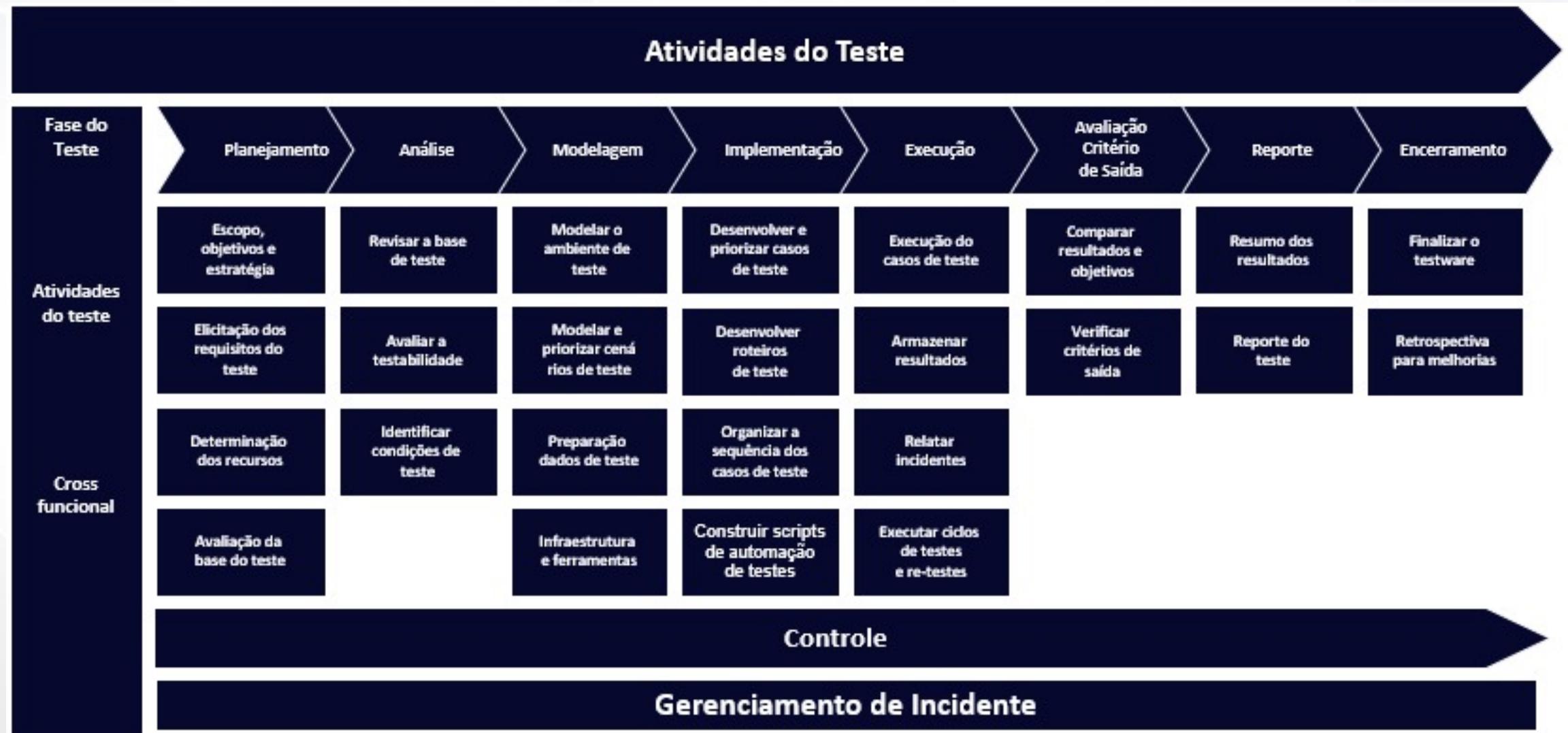




PrimeControl
Quality Drives Results

O PROCESSO DE TESTE

PROCESSO DE TESTE DO ISTQB



PROCESSO DE TESTE



	Planejamento	Análise	Modelagem	Implementação	Execução	Reporte	Encerramento
Atividades	Entendimento da demanda	Entendimento do sistema Inventário do sistema	Cenários Identificação dos Casos de Teste Detalhar funções do sistema	Elaboração dos Casos de Teste	Planejamento dos teste Execução dos testes	Elaborar relatórios	Finalizar testware
Artefatos	Estimativa demanda Estratégia de Teste	Estratégia de Teste (refinamento)	Cenário de Teste Detalha-mento	Caso de Teste	Relatório de execução Relatório de incidentes	Relatório gerenciais	
	Analista Teste	Analista Teste	Analista de Teste	Analista Teste	Executores Analistas de Teste	Analista Teste	Analista Teste

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ENTENDIMENTO DA DEMANDA



- Para iniciar as atividades de teste é necessário que o líder do teste entenda a demanda do teste
- É necessário alinhamento com o cliente
- Todas as necessidades e restrições devem ser esclarecidas para que as atividades de teste possam ser realizadas com sucesso



ENTENDIMENTO DA DEMANDA



- Quais objetivos do teste?
- Qual o escopo do teste?
- Como é o ambiente de teste?
- Quem é o responsável pelo ambiente de teste?
- Quem é responsável pela massa de teste?
- Como será tratada a massa de teste?



ENTENDIMENTO DA DEMANDA



- Quais documentos estarão disponíveis sobre o sistema?
- Como é o contato com o cliente para esclarecimentos sobre o sistema?
- Qual a visão macro do sistema?
- Quantos ciclos de regressão serão realizados?
- Quais ferramentas serão utilizadas?

TÉCNICAS DE ENTENDIMENTO DA DEMANDA



- Possibilidade de checklist
- Como conduzir o entendimento da demanda?
 - Realização de reuniões e entrevistas com os stakeholders e usuários para alinhamento e entendimento do escopo de projeto
- Toda a documentação gerada para entendimento da demanda deve ser armazenada em um repositório para controle de versão e de fácil acesso aos membros envolvidos no projeto

EXEMPLO DE CHECKLIST



Existe um contato inicial?

- Sim Não Nome Contato:

Existe um Ponto Focal de Comunicação

- Sim Não Nome Ponto Focal:

Qual departamento requisitante está interessado no sistema?

Nome Departamento:

Documentação Disponível do Sistema?

- Especificação de Requisitos Caso de Uso Outros

O Sistema é?

- Novo Manutenção Legado

Disponibilidade de usuário/clientes para entrevistas?

- Sim Não

Em caso de Sistema legado, o sistema está disponível para acesso?

- Sim Não

Em caso de Sistema Novo, o sistema está disponível para acesso?

- Sim Não

Em Caso de Sistema para Manutenção, o sistema está disponível para acesso?

- Sim Não

COMO CONDUZIR O ENTENDIMENTO DA DEMANDA?



- Técnicas de Entrevistas

Adequado para	Não é adequado porque
Coletar fatores básicos e de performance (conhecimento explícito)	Os stakeholders devem estar cientes das suas necessidades e poder explicá-las

- Entrevista

Questionamento ao stakeholder apoiada em questões pré-definidas

- Pontos abertos podem ser resolvidos imediatamente

- Questionário

Questões pré-definidas (papel, on-line) para serem respondidas pelos stakeholders

COMO CONDUZIR O ENTENDIMENTO DA DEMANDA?



- Entrevista

Adequado para	Não é adequado porque
Atenção individual ao entrevistado permitindo coletar todos os fatores	Porque consomem tempo Quando a quantidade de tópicos a serem questionados é limitada

- Questionário

Adequado para	Não é adequado porque
Questões são mais amplas Custo e tempo baixos	Captura somente conhecimento explícito Consome tempo de preparo Pode ser combinado com outras técnicas de elicitação

COMO CONDUZIR O ENTENDIMENTO DA DEMANDA?



Workflows	Atividades / Tarefas
Preparar entrevistas	Identificar participantes Definir pauta
↓	Entender o projeto Entender objetivos do projeto Entender status do projeto Entender o plano do projeto Entender metodologia e processo de desenvolvimento Identificar os grandes requisitos
Conduzir a entrevista	Conduzir análise de risco
↓	Resumir entrevistas Confirmar dados das entrevistas
Organizar informações	

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ESTRATÉGIA DE TESTES



- Definição da Abordagem
- Definição do Tipo de Teste
- Definição das Ferramentas
- Definição dos Perfis

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ENTENDIMENTO DO SISTEMA



Utilizando a estratégia de teste, os analistas precisam entender detalhadamente o sistema para viabilizar a modelagem dos testes

- Fluxos
- Requisitos do sistema
- Regras de negócio
- Funcionalidades
- Casos de uso
- Navegação

ENTENDIMENTO DO SISTEMA



Análise de criticidade pode ser realizada para priorização e análise de abrangência dos testes

Não devemos desgastar o relacionamento com o cliente nessa fase. É necessário fazer um estudo prévio criterioso para sermos assertivos nas interações com o cliente.

ENTENDIMENTO DO SISTEMA



O sistema pode ter diferentes situações de documentação

- Sistema bem documentado
- Sistema documentado, porém de difícil utilização
- Sistema mal documentado
- Sistema sem nenhuma documentação

É necessário entender diferentes estratégias para realizar o entendimento do sistema em qualquer uma das situações

ENTENDIMENTO DO SISTEMA



Ao fazermos o entendimento do sistema podemos elaborar alguns documentos de apoio

- Macro visão do sistema
- Fluxos de funcionalidade
- Registros do estudo dos requisitos e documentação existente

Manter documentação sobre o entendimento do sistema auxilia em atividades futuras de manutenção

ENTENDIMENTO DO SISTEMA



Sugestão de atividades para um melhor entendimento do sistema:

- A leitura dos documentos de apoio
- Exploração e/ou observação do sistema – na falta de documentos e disponibilidade do sistema
- Entrevista ou questionário – não havendo documentação e nem disponibilidade do sistema

ENTENDIMENTO DO SISTEMA



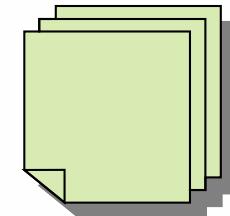
Resumo



checklist e montar questionário



entrevista com cliente



elaboração de documentação de apoio



validação com cliente

Essas etapas não são aplicadas para todos os projetos

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ANÁLISE DE COBERTURA DO TESTE



Com as informações obtidas do sistema o analista de teste deve fazer inicialmente uma modelagem de alto nível dos testes

- Cobertura dos requisitos
- Cenários de teste
- Análise da abrangência dos testes (utilizando riscos)

ANÁLISE DE COBERTURA DO TESTE



Fornecemos uma visão macro dos testes

- O cliente pode analisar e aprovar o que será testado
- Traz entendimento da cobertura do teste
- Os analistas devem utilizar para guiarem a escrita de seus casos de teste

CENÁRIO DE TESTE



- É um planejamento inicial dos casos de teste
- Traz visão da cobertura dos testes
- Um cenário pode conter um ou mais casos de teste
- É importante que sejam validados com o cliente

CENÁRIO DE TESTE



No levantamento dos cenários de teste registra-se as diferentes condições de teste de acordo com as técnicas definidas no documento de estratégia de testes

- Variações possíveis e necessárias para cobrir um determinado requisito
- Para efetuar esta atividade é necessário o conhecimento do negócio – realizado na fase anterior

CENÁRIO DE TESTE



Uma das técnicas para identificar os cenários de teste é baseada em caso de uso

- Cenários positivos
- Cenários de exceção
- Cenários negativos

ETAPAS PARA IMPLEMENTAÇÃO DOS TESTES



Entendimento da
Demanda



Elaboração da
Estratégia de Teste



Entendimento do
Sistema



Análise da
Cobertura do Teste



Elaboração dos
Casos de Teste

ELABORAÇÃO DOS CASOS DE TESTE



- Com os cenários de teste prontos podemos elaborar os casos de teste
- Com os casos de teste cobrimos os requisitos de acordo com os cenários de teste identificados e escopo da demanda

DEFINIÇÃO DE CASO DE TESTE



Um caso de teste deve contemplar um conjunto de valores de entrada/inputs, precondições de execução, resultados esperados e pós-condições de execução desenvolvidas para um determinado objetivo ou condição de teste, tais como para exercitar o caminho de um determinado programa ou verificar o atendimento a um requisito específico.

[IEEE 610]

DIRETIVAS DO CASO DE TESTE



Completude

- um caso de teste deve ser completo em si mesmo, ou seja, não deve depender de nenhum outro caso para que possa ser executado ou para que seu resultado possa ser aferido

Atomicidade

- um caso deve ter o escopo sobre as condições de teste endereçadas reduzido ao máximo, ou seja, deve ser o mais simples possível em termos de abrangência

QUALIDADE DO CASO DE TESTE



O caso de teste deve seguir padrões de qualidade:

- Efetivo: testar o que se planejou testar
- Econômico: sem passos desnecessários
- Reutilizável: possa ser repetido
- Rastreável: possa identificar o requisito a ser testado
- Autoexplicativo: possa ser testado por qualquer pessoa



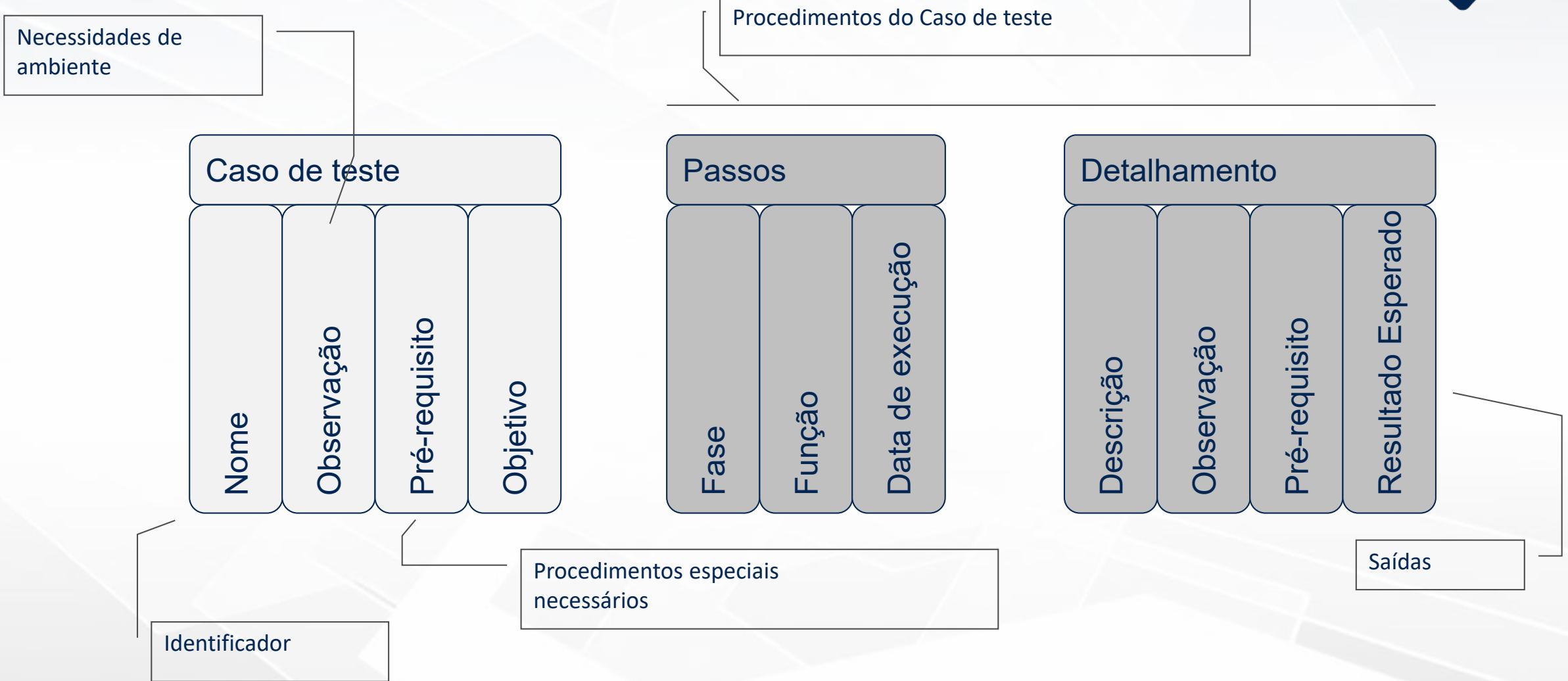
ADERÊNCIA DO MODELO A IEEE 829



O modelo utilizado para a escrita de casos de teste é aderente à IEEE 829, reunindo os documentos de

- Caso de teste
- Procedimento de teste

ADERÊNCIA DO MODELO A IEEE 829



CASO DE TESTE – DOCUMENTO



- Caso de teste
 - Nome: Esse é o nome do Caso de teste
Exemplo: Saque parcial de valor disponível dentro do limite diário
 - Observação: Alguma informação em especial que necessite ser adicionada para o caso de teste
Exemplo: Regras de execução

CASO DE TESTE – DOCUMENTO



- Caso de teste (cont)
 - Pré-Requisito: Atributos que devem ser previamente atendidos para execução do caso de teste

Exemplo: Conta corrente existente com aplicação em fundo disponível para resgate e cartão de débito com senha válida
 - Objetivo: Meta a ser obtida com a execução do caso de teste em questão. Deve ser elaborado em forma de texto explicativo

Exemplo: Realizar saque parcial em conta corrente através de um terminal ATM utilizando cartão de débito. O valor está dentro do limite da transação e o saque é realizado com sucesso

CASO DE TESTE – DOCUMENTO



- Passos
 - Fase: Declaração das execuções de testes de Setup, Teste, Verificação e Clean-up
 - Exemplo: Setup
 - É importante que o Analista de Teste conheça os meios de executar as fases Setup, Verificação e Clean-Up da forma mais econômica possível

CASO DE TESTE – DOCUMENTO



- Entendimento dos Passos
 - Setup: são funções que devem preparar as condições para executar o teste
 - Cuidado, pois setup é diferente de pré-requisito!
 - Teste: são as funções alvo do teste, sendo suas variações inseridas neste contexto

CASO DE TESTE – DOCUMENTO



- Entendimento dos Passos
 - Verificação: são ações que validam o resultado do teste. A verificação pode ocorrer em qualquer área do sistema que comprove se a transação efetuada corresponde ao resultado esperado
 - Exemplo: a verificação se o relatório foi impresso com o layout adequado (procedimento manual)
 - Exemplo: extrato de uma conta para verificar um saque
 - Clean-Up: são funções ou ações que permite retornar à posição inicial (anterior ao teste). Esta fase do caso de teste é opcional, porém para torná-lo mais completo poderá ser considerada

CASO DE TESTE – DOCUMENTO



- Passos (cont)
 - Função: Nome da Função para execução. Dentro da estrutura do RSI/QA-Teste esse nome de função está na estrutura: Sistema\Componente\Função
Exemplo: Resgate de Fundos
Outro exemplo: ATM\Aplicação\ResgateFundos
 - Data de Execução: Data de referência para a execução das funções do caso de teste. Deve ser no formato D+X (onde X é o número de dias que devemos tomar a partir da data de execução)
Exemplo: D+0

CASO DE TESTE – DOCUMENTO



- Detalhamento
 - Descrição: Informação descritiva, em forma de texto, sobre os passos que serão executados

Exemplo: Efetuar Resgate do Fundo na Conta Corrente em valor superior a R\$ 50,00
 - Observação: Detalhamento necessário para a execução dos passos

Exemplo: 1) Com o caixa aberto, clicar em Investimento > Resgate > Fundo 2) Selecionar o Fundo que a conta tem valor disponível para resgate 3) Inserir o valor a ser resgatado, acima de R\$ 50,00

CASO DE TESTE – DOCUMENTO



- Detalhamento (cont)
 - Pré-Requisitos: Atributos que devem ser previamente atendidos para execução dos passos do caso de teste

Exemplo: Conta corrente ativa com Aplicação em Fundo disponível para resgate
 - Resultado Esperado: Situação, condição ou valor desejado com a execução dos passos do caso de teste

Exemplo: Resgate realizado com sucesso com o valor solicitado



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





PrimeControl
Quality Drives Results

TÉCNICAS DE MODELAGEM DE TESTE

TÉCNICAS DE MODELAGEM DE TESTE – CATEGORIAS



- **Baseado na especificação (caixa preta)**
 - Crie testes primariamente pela análise da base de teste
 - Procure por bugs no modo do sistema se comportar
- **Baseado na estrutura (caixa branca)**
 - Crie testes primariamente pela análise da estrutura do componente ou sistema
 - Procure por bugs no modo que o sistema é construído
- **Baseado na experiência (ataques, listas de checagem, teste exploratório)**
 - Crie testes primariamente baseados no entendimento do sistema, experiência anterior, e suposições sobre bugs
 - Procure por bugs nos lugares onde os outros sistemas apresentam bugs
- **Testes de caixa preta são baseados em especificação ou experiência, funcional ou não-funcional.**



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





PrimeControl
Quality Drives Results

TÉCNICAS DE MODELAGEM DE TESTE – CAIXA PRETA

ELABORAÇÃO DE CASO DE TESTE



Conjunto de valores de entrada, precondições de execução, resultados esperados e pós-condições de execução desenvolvidas para um determinado objetivo ou condição de teste, tais como para exercitar o caminho de um determinado programa ou verificar o atendimento a um requisito específico.

[IEEE 610]

PARTIÇÃO DE EQUIVALÊNCIA



- Parte de uma entrada ou saída de domínio para o qual o comportamento de um componente ou sistema é assumido ser o mesmo, baseado na especificação.
- Divide as entradas, saídas, comportamentos, e ambientes em classes que você acredita que serão tratadas de forma equivalente
- Define pelo menos um caso de teste para cada partição ou usa valores limites nas partições em que há uma variação

PARTIÇÃO DE EQUIVALÊNCIA



Entrada	Valores Permitidos	Classes	Casos de Teste
Idade (este valor é obtido através da digitação da data de aniversário)	Número entre 18 a 120	18 a 120	Idade = 20
		< 18	Idade = 10
		>120	Idade = 150

ANÁLISE DE VALOR LIMITE



- O comportamento nos limites de uma partição de equivalência é onde existe maior probabilidade de estar incorreto.
- Limites são áreas onde testes estão mais propensos a indicar defeitos.
- Os valores limites de uma partição são seu máximo e seu mínimo.
- Deve ser usada quando os elementos da partição de equivalência são ordenados.

ANÁLISE DE VALOR LIMITE



Entrada	Valores Permitidos	Classes	Casos de Teste
Idade (este valor é obtido através da digitação da data de aniversário)	Número entre 18 a 120	18 a 120	Idade = 18 Idade = 120
		< 18	Idade = 17
		>120	Idade = 121

EXERCÍCIO



Um campo aceita como entrada um inteiro entre 1 e 99 representando a quantidade de um item a ser comprado. Quais seriam os testes utilizando Partição de Equivalência e Análise de Valor Limite?

EXERCÍCIO



Um programa valida campos numéricos da seguinte maneira:

- Valores menores do que 10 são rejeitados;
- Valores entre 10 e 21 são aceitos;
- Valores maiores ou iguais a 22 são rejeitados

Qual dos seguintes valores de entrada cobre todas as partições de equivalência?

- 1) 10, 11, 15, 21
- 2) 3, 20, 21
- 3) 3, 15, 23
- 4) 10, 21, 22, 23

EXERCÍCIO



Usando as mesmas especificações da questão anterior, qual dos seguintes valores de entrada cobre a maioria dos valores limites?

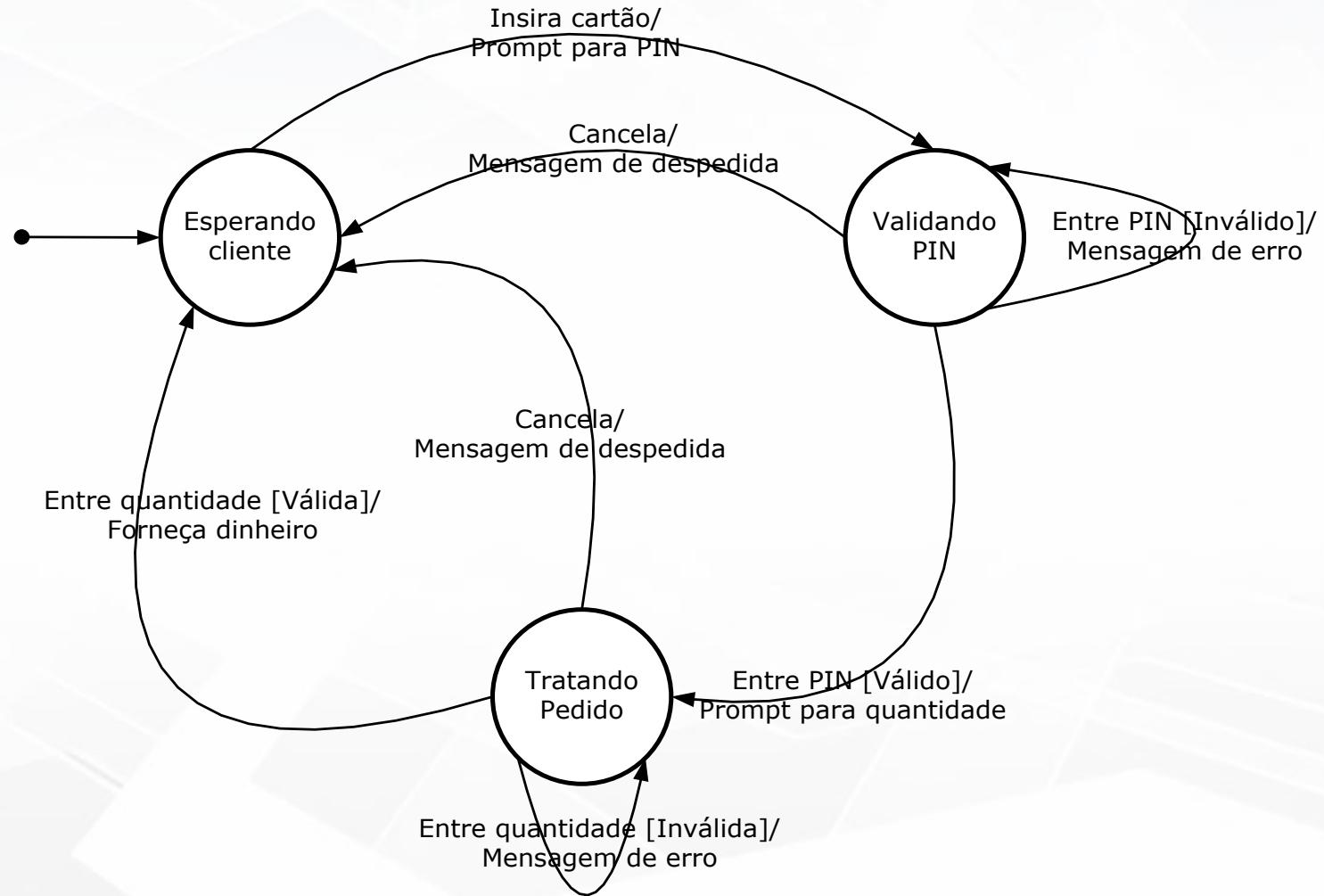
- 1) 9, 10, 11, 22
- 2) 9, 10, 21, 22
- 3) 10, 11, 21, 22
- 4) 10, 11, 20, 21

TRANSIÇÃO DE ESTADO



- Um sistema pode exibir respostas diferentes dependendo da sua condição atual ou de estado anterior.
- O comportamento do sistema pode ser representado como um diagrama de transição de estados.
- Permite ao testador visualizar o software em termos de estados, transições entre estados, as entradas ou eventos que disparam as mudanças de estado (transição) e as ações que podem resultar daquelas transições.

TRANSIÇÃO DE ESTADO



TRANSIÇÃO DE ESTADO

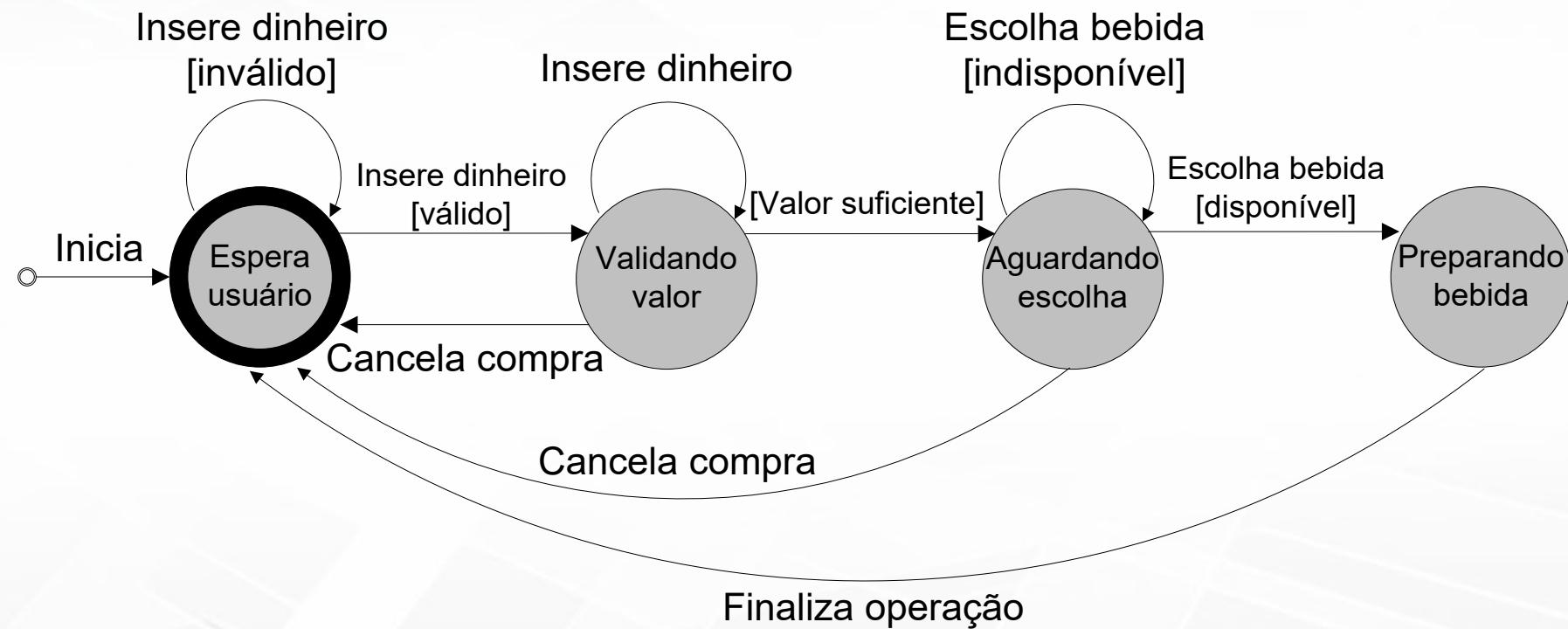


TABELA DE DECISÃO



- Tabela que mostra as combinações de entradas e/ou estímulos (causas) com suas saídas e/ou ações (efeitos) associadas, que podem ser utilizadas para projetar casos de testes.
- Elas podem ser mostradas como gráficos de fluxo ou como tabelas
- As tabelas de decisão podem produzir casos de teste instantâneos

TABELA DE DECISÃO

Esta tabela de decisão mostra a lógica de negócio para um ATM. Note que os traços “-” indicam condições que não serão atingidas como parte desta regra.

As regras são mutuamente exclusivas, e somente aquela única regra pode ser aplicada em qualquer momento único de tempo.



<i>Condição</i>	1	2	3	4	5
Cartão válido	N	S	S	S	S
Senha válida	-	N	N	S	S
Senha inválida=3	-	N	S	N	N
Saldo OK	-	-	-	N	S

<i>Ação</i>					
Rejeita cartão	S	N	N	N	N
Redigita senha	N	S	N	N	N
Prende cartão	N	N	S	N	N
Redigita opção	N	N	N	S	N
Fornece dinheiro	N	N	N	N	S

BASEADO EM CASO DE USO



- Testes podem ser especificados a partir de casos de uso ou cenários de negócios.
- Caso de Uso descreve interações entre atores (usuários e sistema) produzindo um resultado relevante para um usuário do sistema.
- O Caso de Uso tem pré-condições, que precisam ser garantidas para que o caso de uso funcione.
- O Caso de Uso é finalizado com uma pós-condição que representa os resultados observados e o estado final do sistema após o término do Caso de Uso.

BASEADO EM CASO DE USO



Realizar
Pedidos

Sistema de Vendas

Fluxo Básico

- Cliente quer realizar um pedido

Fluxos Alternativos

- Cliente quer mudar a quantidade de um produto
- Cliente quer cancelar um pedido
- Cliente quer excluir um item de um pedido
- Cliente quer mudar as condições de fornecimento do pedido
- Cliente quer reservar um pedido
- Cliente quer estender a garantia de um item de pedido

Fluxos de Exceção

- Cliente quer realizar um pedido, mas seu cartão está inválido
- Cliente quer realizar um pedido, mas possui problemas com cheques devolvidos
- Cliente quer realizar um pedido, mas não tem estoque

ELABORAÇÃO DE CASO DE TESTE



- Com os cenários de teste prontos podemos elaborar os casos de teste
- Com os casos de teste cobrimos os requisitos de acordo com os cenários de teste identificados e escopo da demanda



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





PrimeControl
Quality Drives Results

**TÉCNICAS DE MODELAGEM DE TESTE –
BASEADO NA EXPERIÊNCIA**

TÉCNICAS BASEADAS NA EXPERIÊNCIA



- Testes baseados na experiência são baseados no testador...
 - ... habilidade e intuição.
 - ... experiência com aplicações similares.
 - ... experiência com tecnologias similares.
- Ao invés de serem pré-projetados, os testes baseados na experiência são geralmente criados durante a execução do teste (isto é, a abordagem de teste é dinâmica).
- Testes são frequentemente limitados no tempo ou “time-boxed” (isto é, breves períodos focados em condições de teste específicas).
- Exemplos incluem suposição de erros, caça ao bug, “quebra do software” baseada em listas de checagem ou taxonomias dos bugs, e teste exploratório.

TÉCNICAS BASEADAS NA EXPERIÊNCIA



- Muitas abordagens profissionais para teste baseado na experiência não criam testes totalmente durante a execução.
- Podem ser guiadas por:
 - Lista de checagem
 - Taxonomia do bug
 - Lista de ataques
 - Abordagem de caça ao bug
 - Conjunto de gráficos de teste
- Estas linhas mestras são preparadas antes do teste com algum nível de detalhe.
- Teste puramente “on-the-fly” (teste ad hoc) é comum, mas trata-se usualmente de (ineficiente) teste manual aleatório.

TÉCNICAS BASEADAS NA EXPERIÊNCIA



Vantagens

- Efetivo na descoberta de *bugs*
- Resiste ao paradoxo do pesticida devido à grande variedade
- Eficiente (mantém registros pequenos)
- Boa checagem nos testes preparados
- Divertido e criativo

Desvantagens

- Cobertura com lacunas, especialmente quando sob pressão
- Difícil de estimar
- Sem prevenção de *bug*
- Discussão extensiva não é adequada a grandes equipes
- Nem todos os testadores têm o nível necessário de habilidade e experiência



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA





PrimeControl
Quality Drives Results

Cultura Ágil

Manifesto Ágil



- ❖ Declaração de valores e princípios essenciais para o desenvolvimento de software;
- ❖ Criado em fevereiro de 2001, por 17 profissionais que praticavam métodos como XP, DSDM, SCRUM e FDD;
- ❖ Corresponde aos pontos comuns de projetos que tiveram sucesso em suas metodologias.



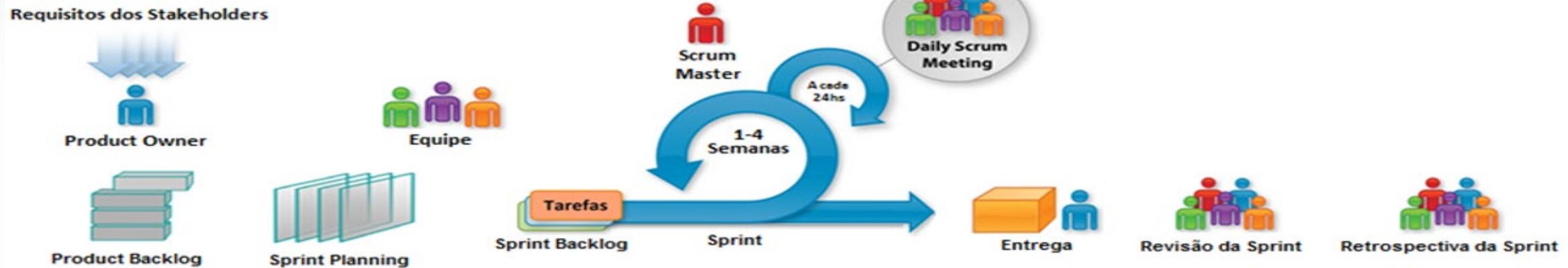


Manifesto Ágil - Valores

- ❖ Indivíduos e interações mais que processos e ferramentas;
- ❖ Software em funcionamento mais que documentação abrangente;
- ❖ Colaboração com o cliente mais que negociação de contratos;
- ❖ Responder a mudanças mais que seguir um plano.

- ✓ Mesmo havendo valor aos itens à direita, valorizamos mais os itens à esquerda.
- ✓ Acesse o site: <http://www.manifestoagil.com.br>

Scrum



- **Características:**

- Iterativo e Incremental
- Colaborativo
- Adaptativo

- **Benefícios:**

- Estimativa de tempo e custo mais precisas
- Aumento da visibilidade e controle do processo
- Aumento da qualidade: testes a cada sprint
- Uma entrega a cada Sprint

Cultura Ágil

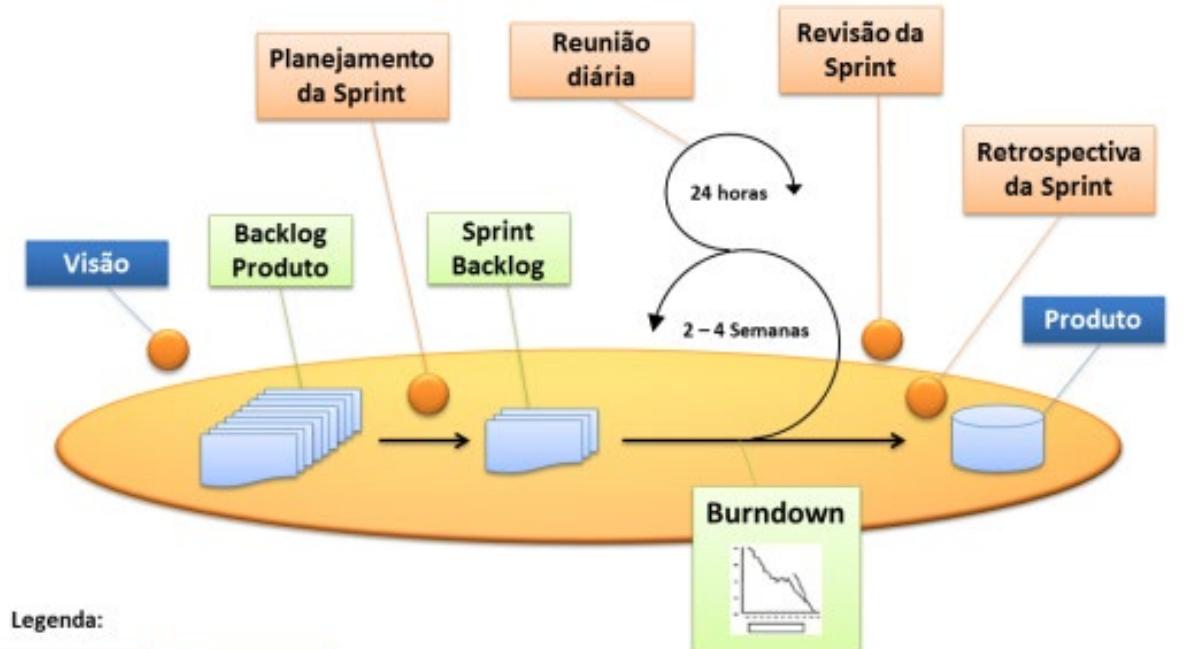


Metodologia Ágil Scrum:

- ❖ Prioriza a entrega do software;
- ❖ Documentação sucinta;
- ❖ Controle das atividades rotineiras.

4 Artefatos fundamentais

- ❖ Product Backlog;
- ❖ User Story;
- ❖ Sprint Backlog;
- ❖ Gráfico de Burn Down.



Legenda:

Cerimônia

Artefatos

Papéis

- Product Owner (PO)
- Scrum Master (SM)
- Equipe Scrum

Cerimônias

- Planejamento da Sprint
- Reunião diária
- Revisão da Sprint
- Retrospectiva da Sprint

Artefatos

- Product Backlog
- Sprint Backlog
- Burndown (gráfico)

SCALt
ENDEAVOR UP

*ISG Provider Lens®
Quadrant Study

2020
Leading Quadrant

*ISG Provider Lens®
Quadrant Study

2020
Leading Quadrant





PrimeControl
Quality Drives Results

FERRAMENTAS DE TESTE

ATIVIDADES SUPORTADAS POR FERRAMENTAS



- As ferramentas dão suporte a várias atividades de teste
- Suporte direto
- Suporte gerencial
- Suporte de monitoração
- Ferramentas básicas

OBJETIVO DAS FERRAMENTAS DE TESTE



- As ferramentas de teste serve a vários objetivos
- As ferramentas podem aumentar a eficiência do teste
- As ferramentas podem automatizar atividades manuais difíceis
- As ferramentas podem automatizar atividades impossíveis de serem executadas manualmente
- As ferramentas podem aumentar a confiabilidade do teste

CLASSIFICAÇÃO DAS FERRAMENTAS DE TESTE



- Muitas ferramentas estão disponíveis para o teste
- O ISTQB classifica ferramentas por atividade de teste
- Existem ferramentas comerciais, freeware, shareware e de código aberto. Muitas opções disponíveis
- Ferramentas são invasivas ou não-invasivas:
 - Ferramentas invasivas têm efeitos investigativos
 - Ferramentas não-invasivas são normalmente mais caras
- Algumas ferramentas são primariamente para desenvolvedores
- O termo “framework de teste” pode se referir a várias coisas, mas o ISTQB utiliza o termo para abordagem de modelagem de automação ou equipamentos de teste

FERRAMENTAS DE GERENCIAMENTO DE TESTE



- Fornece rastreabilidade dos testes, resultados dos testes e incidentes para a base de teste
- Permite o registro de resultados de teste, e geração de relatórios de progresso
- Auxilia a gerenciar os testes e o processo de teste
- Interface para a execução do teste, rastreamento de bugs, e ferramentas de gerenciamento de requisitos
- Fornece controle de versão ou interface com uma ferramenta externa de gerenciamento de configuração
- Executa análise quantitativa (métricas) relacionada aos testes

FERRAMENTAS DE GERENCIAMENTO DE REQUISITOS



- Armazena declarações de requisitos
- Verifica consistência e requisitos indefinidos (faltantes)
- Permite que os requisitos sejam priorizados
- Permite que testes individuais sejam rastreáveis para (e em termos de) requisitos, funções, e/ou funcionalidades

FERRAMENTA DE GESTÃO DE BUGS (DEFEITOS OU INCIDENTES)



- Armazena e gerencia relatórios de bug
- Facilita a priorização / classificação do bug
- Provê workflow baseado em estados, incluindo atribuição de ações para pessoas (por exemplo, correção, teste de confirmação, etc.)
- Permite monitoramento dos bugs de um projeto e status do bug ao longo do tempo
- Provê apoio para análise estatística (geralmente através de exportação; por exemplo, para o Excel)
- Cria relatórios e gráficos (apesar de geralmente ser de utilidade limitada devido à variação nas necessidades)

FERRAMENTAS DE GESTÃO DE CONFIGURAÇÃO



- Armazena informação sobre versões e builds do software e testware
- Permite rastreabilidade entre testware e produtos do trabalho de software e variantes do produto
- Ajuda no desenvolvimento e no teste em múltiplas configurações de ambientes de hardware/software

FERRAMENTAS DE APOIO AO PROCESSO DE REVISÃO



- Armazena informação sobre processos de revisão
- Armazena e comunica comentários de revisão
- Relato de defeito e esforço
- Gerencia referências para regras de revisão e/ou lista de checagem
- Provê ajuda para revisões online
- Viabiliza rastreabilidade entre documentos e código fonte

FERRAMENTAS DE ANÁLISE ESTÁTICA



- Primariamente usadas por desenvolvedores
- Encontra defeitos antes do teste dinâmico
- Força uso de padrões de codificação
- Analisa estruturas e dependências
- Ajuda no entendimento do código fonte
- Calcula métricas a partir do código

FERRAMENTAS DE MODELAGEM E PROJETO



- Primariamente usadas por desenvolvedores
- Ajuda a criar modelos do sistema
- Valida modelos
- Ajuda na geração de alguns casos de teste baseados no modelo

FERRAMENTA DE MODELAGEM DE TESTE



- Gera entradas de teste ou testes reais a partir de:
 - Requisitos
 - Interface Gráfica com Usuário
 - Projeto de modelos
 - Código
- Gera resultados esperados (apesar de que a confiabilidade desse tipo de oráculos de teste ser frequentemente limitada)
- Gera *frameworks* de teste, modelos e *stubs*

FERRAMENTAS DE DADOS DE TESTE



- Manipula ou cria banco de dados, arquivos ou dados para uso durante a execução do teste
- Cria grandes volumes de dados de teste úteis
- Valida dados de teste de acordo com regras específicas
- Analisa os dados para frequência de condições, etc.
- Mistura ou torna anônimos dados reais ou de clientes

FERRAMENTAS DE EXECUÇÃO DE TESTES



- Executa testes (isto é, submete entradas por um *script* automatizado e compara resultados obtidos e esperados)
 - Usado apropriadamente, o *script* é um procedimento de teste independente da entrada que permite a você repetir os testes com dados diferentes, executar testes similares, e facilmente manter os testes
 - Usados inapropriadamente – por exemplo, típica *captura-playback* – os *scripts* são misturados com os dados e/ou resultados esperados, possuem atrasos codificados diretamente neles, e são muito frágeis
 - *Captura-playback* pode ser útil durante o teste exploratório
- Inclui comparadores
- Produz registros analisáveis
- Trabalha na GUI, API ou CLI.

FRAMEWORKS DE TESTE, AMBIENTE PREPARADO E SIMULADORES



- Trocar ou substituir peças de hardware faltantes e/ou potencialmente problemáticas
- Facilitar o teste de unidade(s) gerando e / ou suportando *drivers*, *stubs*, e/ou objetos simulados que substituem porções do sistema, que são indisponíveis ou removidos para isolar a unidade
- Prover *frameworks* de execução no *middleware* para testar linguagens, sistemas operacionais ou hardware

COMPARADORES DE TESTE



- Checa arquivos, banco de dados, ou resultados do teste mediante expectativas ou durante a execução do teste ou subsequentemente
- Pode incluir um oráculo de teste automatizado ao invés de comparar mediante linhas de base estáticas
- Pode ser tornado esperto através da sua programação para tratar variáveis dinâmicas como datas e tempo, ou ignorar a ordem dos registros quando a ordem de seleção em relatórios ou consultas é ambígua

FERRAMENTAS DE MEDIDA DE COBERTURA



- Primariamente usadas por desenvolvedores
- Tanto invasiva como não-invasiva
- Mede a porcentagem de tipos específicos de estrutura de código que foram exercitados:
 - Comandos
 - Desvios ou decisões
 - Objetos
 - Chamadas de função
- Checa quanto fundo um conjunto de testes executou o tipo de estrutura medido

FERRAMENTAS DE SEGURANÇA



- Checa por vírus de computador
- Simula várias formas de ataques
- Simula várias condições de segurança
- Checa o código para procurar violações de segurança

DESEMPENHO, MONITORAMENTO E ANÁLISE DINÂMICA



- Primariamente usadas por desenvolvedores, apesar de muitos testadores usarem ferramentas de desempenho e monitoramento
- Ferramentas de análise dinâmica frequentemente são usadas para checar por dependências de tempo ou não liberação de memória
- Monitora e relata como um sistema se comporta sob condições de uso simuladas
- Gera várias (esperançosamente realistas) condições de carga para a aplicação, um banco de dados, rede ou servidor, geralmente por algum script ou procedimento programado
- Monitora, analisa, verifica e relata o uso de recursos específicos do sistema, e fornece avisos de problemas possíveis

OUTRAS FERRAMENTAS



- Algumas ferramentas são focadas em aplicações particulares
 - Ferramentas de teste de desempenho baseada na *Web*
 - Ferramentas de análise estática específicas da linguagem
 - Ferramentas de teste de segurança
- Algumas áreas alvo específicas da aplicação como sistemas embarcados
- É claro, testadores também usam planilhas (muitas delas!) e banco de dados
- Muitos desenvolvedores usam ferramentas de depuração



E AÍ PESSOAL, TEMOS
ALGUMA DÚVIDA



SER PROTAGONISTA PARA GERAR VALOR NO NEGÓCIO DOS NOSSOS CLIENTES.



PrimeControl

Quality Drives Results



Empresa participante do
SCALE ENDEAVOR UP

